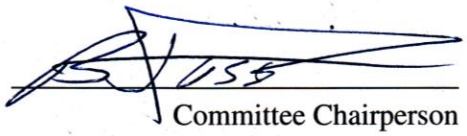
Development of a Programmable Integrated Switch Matrix (PriSM)

A THESIS

APPROVED FOR THE DEPARTMENT OF ENGINEERING AND PHYSICS



Committee Chairperson

Committee Member

Committee Member

UNIVERSITY OF CENTRAL OKLAHOMA

Edmond, Oklahoma

Jackson School of Graduate Studies and Research

DEVELOPMENT OF A PROGRAMMABLE INTEGRATED SWITCH MATRIX (PRISM)

A THESIS

SUBMITTED TO THE GRADUATE COLLEGE

In the fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN ENGINEERING PHYSICS

By

Igor Ilikj

Edmond, Oklahoma

2017

Acknowledgements

I would like to extend my gratitude to my advisor for this project, Dr. Baha Jassemnejad, for his support and guidance during my time at UCO. He served as the committee chair for my thesis while he was a part of the Department of Engineering and Physics and continued in his role from his position at the Federal Aviation Administration. I would also like to thank Dr. Ronald Miller and Dr. Alaeddin Abuabed who served on my thesis committee. I would also like to mention the help that I received from Jonathan Adams and Christopher Scott, fellow graduate students who were working related thesis projects. We used knowledge obtained from our own projects and collaborated on multiple occasions. Additionally, I would like to thank the staff at the Engineering and Physics Department in UCO for their help. Last, but not least, I would like to extend my thanks to the engineers working for the OKCET at the FAA, chiefly Mr. Clint Quisenberry and Mr. Barry Foister. They were instrumental in getting this project off the drawing board as well as in the concluding phases.

Table of Contents

Abstract

This research project involved the development of a fully customizable, user-defined hardware-software suite for automated signal routing with function expandability. Because of the above-mentioned characteristics, the project was called Programmable Integrated Switch Matrix (PrISM). This intelligent switching system can be customized and employed in any industry where there is a need for programmable, timed, and/or simultaneous routing of analog or digital signals between devices. Potential applications of these automated switching systems include, but are not limited to: demarcation points, test floors, redundant backup systems, remote maintenance, etc. A suitable test bed for PrISM was found in a collaboration with the Federal Aviation Administration (FAA) Oklahoma Communications Engineering Team (OKCET) Laboratory and has found an immediate potential application as a large-scale switching system. The fundamental hardware unit for this system is the National Instruments (NI) PXI chassis with a NI SwitchBlock populated with matrix relay cards. The chassis can be deployed in any location, contributing to the robust nature of the design. The advantage of using an integrated NI system is its modularity; the hardware can be easily tailored to the specific needs of each end user. Expansion and customization is accomplished with the addition of a wide spectrum of matrix relay cards. Matrix cards are available with a varying number of relays or switching points. The proposed system is controlled and automated by a customized virtual instrument (VI) application software that was developed using NI LabVIEW software environment and can be integrated with the PXI or function as an executable on a standalone desktop computer.

I Introduction

Large-scale switching is performed at facilities that utilize multiple communication/signal/test devices that must frequently be connected, disconnected, and rerouted. This facility could be a massive demarcation point at an airport, responsible for routing vital communications to and from in-flight aircraft; a testing facility or laboratory that must perform large quantities of tests on various devices and/or signals; or any communication hub with large-scale routing needs. A system that performs this large-scale switching needed to be subjected to some requirements in order to have the full functionality required for a specific application. Such requirements range from power specifications on the switch relays and expandability options, to aesthetics of the graphical user interface and bulkiness of the hardware setup.

This thesis proposes the development of a programmable and modular integrated switch matrix. These characteristics are reflected in the name of the project, PrISM (Programmable Integrated Switch Matrix). PrISM needs to be versatile enough to be deployed in different switching applications. Additionally, different applications can require different functionalities as well as vastly different switch matrix size. This can be achieved by designing a system using a modular approach, where functionality is added as required for each specific application. Designing a modular project requires the development of a hardware and software framework that supports plug and play functionalities. This versatility characteristic defines the programmable and integrated approach PrISM stands for.

Although PrISM was designed to be used in any application, it has an immediate potential application in conjunction with the Federal Aviation Administration (FAA) Oklahoma

Communications Engineering Team (OKCET) Laboratory. The lab has a wide range of communication devices and voice switch positions that need to be capable of interfacing with each other. Having a large number of connection points requires a large number of copper wire connections. These hardwire connections need to be routed in an optimal setup to achieve the required functionality while operating at a high capacity in a minimum of space. As seen on Figure II.1, all the devices set in the lab are connected to a large demarcation point. Copper wire connections should be routed from the demarcation point to a switch matrix that can intelligently connect user-defined devices together or connect them to phone lines that access devices with the outside world. The example on Figure II.1 shows two different voice switch positions connected via the switch matrix shown at the bottom of the diagram. This connection is marked in red, connecting voice switch position 1 radio 1 transmit to voice switch position 2 radio 2 receive.

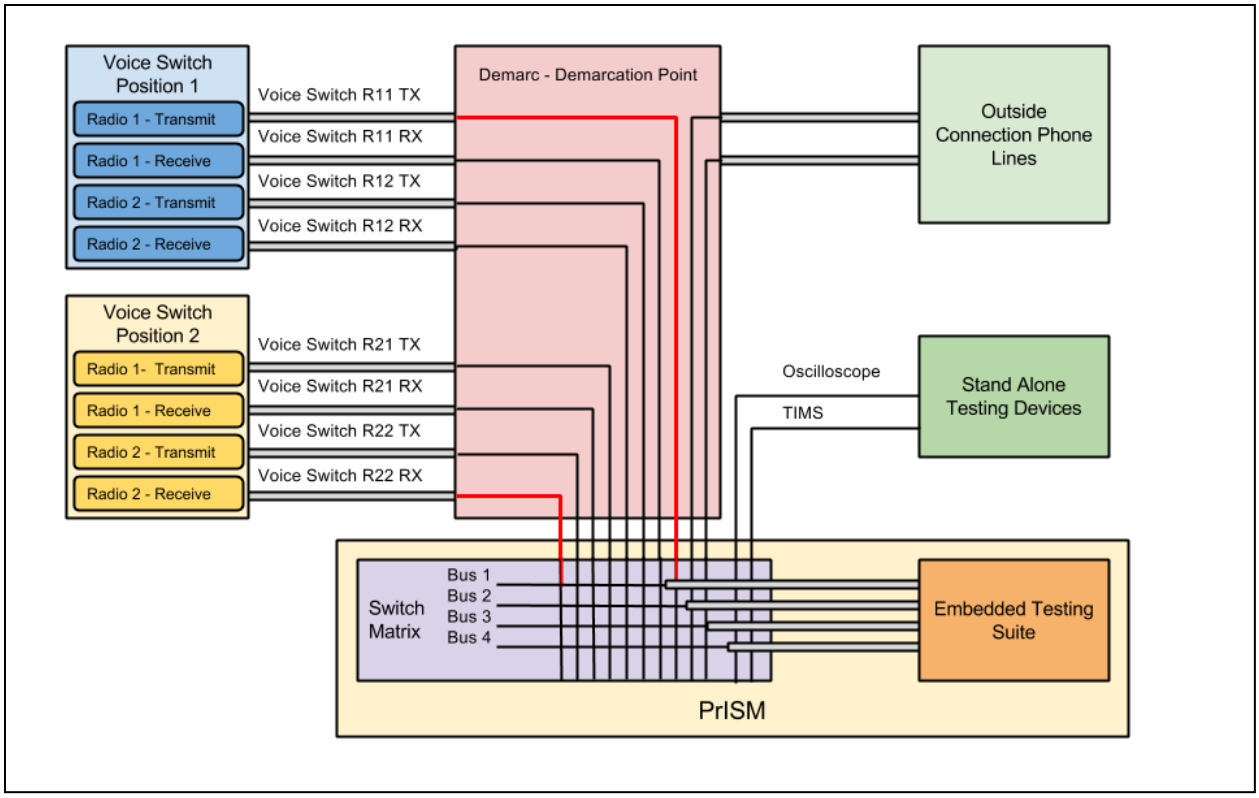


Figure II.1 Hardware Overview

The OKCET laboratory provided an opportunity to use the PRISM functioning as a switch matrix capable of routing between a large number of devices with the possibility of expansion into the thousands. The lab required the ability to achieve multiple simultaneous connections for testing different pieces of equipment at the same time. To fulfil these requirements, in an A x B matrix, the number of rows, A, is a smaller number representing the number of simultaneous connections allowed; whereas the number of columns B, will be a larger number representing the number of devices that would be able to interface with each other. In the case at hand, A can be a number such as 8 or 16, while B needs to be able to grow into the thousands because of the number of devices the laboratory wants to have plugged in the switch matrix. Therefore, the

hardware chosen to accomplish the task must have the ability to be easily expanded so it will be able to meet any future needs that the laboratory might have.

II Objective

II.1 Introduction

The complete development of a fully customizable, user-defined hardware-software suite for automated signal routing with an open-ended functionality profile required choosing the appropriate platform. Additionally, since PrISM has found an immediate application for proof of concept testing, there was a need for clear hardware and software benchmarks. The project was a collaboration with industry representatives. Therefore, to successfully accomplish the laid-out objectives, the industry representatives needed to be satisfied with the results as well.

The software used to incorporate the logic was required to perform intelligent switching and to control the hardware that would be suited for the task as well as any potential hardware for another functionality. Making this match was important, and set the tone for the entire project. Both the hardware and software had to be flexible to allow for expandability in terms of size and function, and have the ability to work in concert with each other. Due to their variety of hardware, and the ease of use of their software, National Instruments (NI) products were chosen for this project. NI hardware is intrinsically controlled by LabVIEW software, which is unique in its software development capabilities and is compatible with all NI products. These products include a wide variety of communication and electrical engineering related equipment. This was very important to achieve the objective of building a future-proof system, with the capability to accept other hardware if necessary.

II.2 Hardware Objectives

The NI hardware chosen for the switch matrix was the PXI-2800 SwitchBlock Carrier. The SwitchBlock can carry matrix relay cards with an array of switching relays which can be arranged in a large switch matrix. The PXI-2800 SwitchBlock is shown in Figure II.2, with all six of the card slots filled with matrix relay cards.

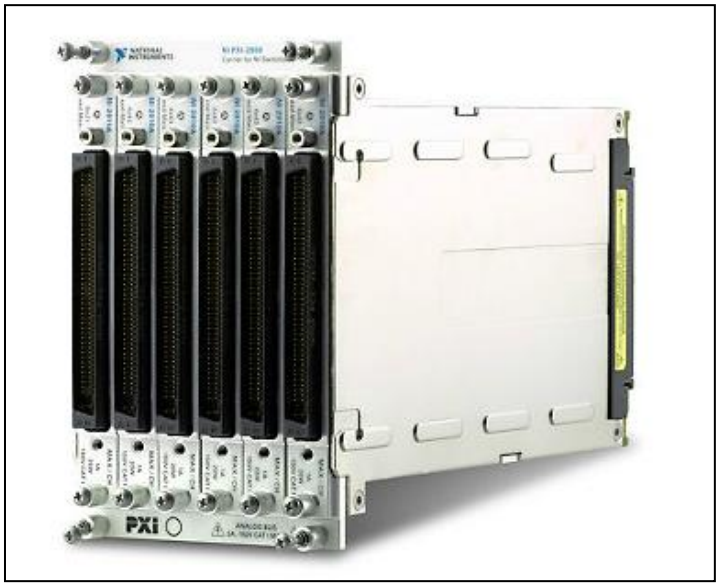


Figure II.2 PXI-2800

The base for a PXI system is a PXI chassis, which can vary in the number of slots available. For a large system, the largest possible chassis is recommended, up to 18 slots. These slots can be taken up by different modules, depending on the need of the system. The SwitchBlock takes up 4 slots on a PXI chassis, and multiple SwitchBlocks can be connected together. Each SwitchBlock has 6 slots that can be occupied by matrix relay cards. The matrix relay cards also have a variety of options to choose from. There are multiple resources from NI on building large scale matrices

[1]. Since the system requires a connection to multiple devices, the expansion plan was laid out in a column expanding configuration as pictured in Figure II.3, shown in the white paper Matrix Expansion Guide by NI [2].

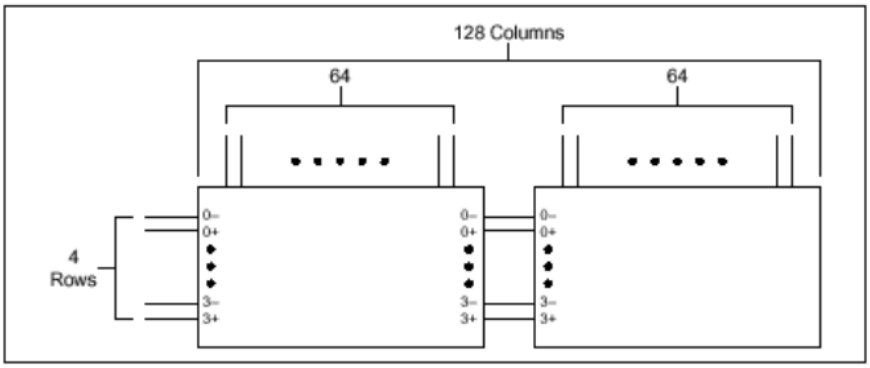


Figure II.3. Column Expansion

There are multiple considerations that must be taken when choosing the type of relays and the switching logic in a switch matrix [3] [4] [5]. NI has an array of switch matrix cards with reed relays that serve as switching points [6]. Reed relays are commonly used as switching points for switch matrices [7] [8]. The other available option is electromechanical relays, which have their advantages in higher current and power allowance, while occupying more space than the reed relays [9] [10]. After considering the previously mentioned white papers from NI [1] [2], as well as consulting with the industry end user for the planned product, a decision was made to go with the electromechanical relays over the reed relays. In this first application of PrISM, the OKCET laboratory had the capability of using PRISM to match the specifications and functionality of their current switch matrix system. One of these specifications is an allowed switching current of up to 2 A. This higher current specification is only available with the bulkier electromechanical relays. A possibility for a larger system would be to mix and match low current devices on reed

relay matrices, and higher current devices on electromechanical relay matrices. In this way, PrISM could utilize the most optimal space-saving setup. The drawback is the inability to connect these two together, because the reed relays can't handle currents up to 2A.

There are two different electromechanical relays available that are SwitchBlock compatible: 4x71 and 8x34. The 8x34 card was chosen, because it allows for at least 8 simultaneous connections at any time. The predicted load of the OKCET laboratory end users would likely require more than 4 simultaneous connections at any given time. These specifications are listed in Table II.1.

Hardware Specifications - NI 2834		
	Prototype	Projected Use
Number of Device Connections	34	1000+
Number of Bus Lines	8	8
Max Switching Current	2.0 A	2.0 A
Max Switching Power	60 W	60 W
Switch Relay Type	EM	EM

Table II.1. NI 2834 Hardware Specifications

II.3 Software Objectives

A set of switch requirements was created to meet the software benchmarks for PrISM. These requirements will govern the development of an algorithm to run this system. Additionally, setting up a palette of features for the industry end user is also important for the success of the proposed system. These include, but are not restricted to, a login system for an administrator and users, a screen to display active connections and possible connections to be made, help menu, troubleshooting guide, and report generation. These requirements are listed in Table II.2.

Algorithm Software Specifications - Prototype	
Device Choice	• Graphical (selecting the devices from images that are grouped by device and signal type). • Text tree (selecting the devices from a list forming a text tree where each branch is a device or signal type).
Switch Relay Positions	• Graphical (pictorial representations of current configuration, showing devices and buses in use). • Text/Table (tabular representation of the switch configuration, listing the devices for each connection and the bus line they occupy).
Routing	• Intelligent routing where the algorithm chooses the first free bus for every new connection. • Options for locking down buses or latching another device on an existing connection. • All buses active notification.
Programmable Connections	• Timed connect/disconnect • Triggered connect/disconnect
Log	• Create a log of connections made • Report generation on user command
Purge	Administrator only purge all connections functionality

Table II.2. Software Specifications

II.4 Collaboration Objectives

This project was developed as a collaboration between industry and academia. Such collaborative ventures are beneficial to both parties. The university parties acquire skill sets that are valuable for their future endeavors beyond their education. Industry representatives are able to harness the power of shared resources with members from the university to achieve industry-oriented goals. Such collaborations have been successfully accomplished via a wide array of university-industry interactions [11] [12] [13] [14]. While these collaboration efforts do open new opportunities, the afore-mentioned "costs" and "benefits" to these collaborations need to be taken in consideration to optimize the interaction [15] [16].

The modular aspect of PrISM allows for future expansion of the available functionalities for the end user. A successful thesis project could ensure that the industry representatives turn to the same academic resources for future projects as well. Streamlining the future of an open-ended project has been an objective of other university-industry collaborations [17]. In turn, this would widen the range of accessible projects, allowing future students to learn more skills in a broader array of fields. This has been done at the undergraduate as well as at the graduate level such as this thesis. Similar graduate student joint collaborations have been accomplished successfully in previous studies [12] [18].

In order to complete the requirements set for PrISM, the system should match the functionality of the OKCET laboratory's current outdated switch matrix, and have the ability to receive future modifications; such as adding more devices or adding completely new functionalities. Therefore, the hardware selected must be able to incorporate the variety of modular pieces capable of

performing other tasks for the industry end user. PrISM allows for additional functionality to be explored by another undergraduate or graduate body of work. Developing a system with this intrinsic ability is the final goal in this modular setup. As shown in other studies, diminishing these orientation and transaction-based gaps has been key in developing university-industry programs [19].

III Hardware Simulation

The first step in determining whether the correct hardware platform was chosen was to perform a hardware simulation. NI provides the capability to simulate the hardware before purchasing any components. This ensures that the hardware is compatible and tested prior to acquisition. Demonstrating the platform effectiveness helps with getting approval for funds to purchase a prototype. Simulation with NI products is done in National Instruments Measurement Automation Explorer (NI MAX). The service provides a simple way to check the integrity of a designed system, and allows for a test of its full functionality [20] [21] [22].

The system will perform large-scale switching with a varying number of devices connected to the matrix. A flexible design allows for the end user to change the parameters of the system. The primary parameter is the number of devices connected, which determines the number of physical connections that need to be in place to handle every device that will be connected to the grid. ‘A’ will be the number of devices that need to connect to the matrix, and ‘B’ would be the number of simultaneous connections available [3] [4] [5]. In the first case of a matrix AxB (A is rows, B is columns), the design would represent a fixed group of devices, A, to be connected to a different set of devices, B. In this case, the rows would be filled in with devices from group A and columns would be devices from group B. A and B would be the number of devices in each group. In another case, all the devices can be connected to each other at any given time. This is also handled by AxB matrix, with the columns being a different device, and the rows representing the multiple connections active at the same time. Communication with the industry representatives for this system is paramount for designing the specifications of the switch matrix. As specified by the NI expansion guide [2], the alignment on Figure II.3 was chosen. The matrix

that will grow in terms of how many devices are connected together by growing in one dimension would be preferable over having to grow the matrix in both dimensions. As described by NI, this would allow for a plug and play expansion by adding more cards on the same bus line [2].

As specified in the objective section, the immediate application of PrISM required a switch matrix with the capability of expanding the number of columns into the thousands. Another aspect that had to be kept in mind was the possibility of adding other functionalities, such as signal generation, measurement, power capabilities, etc. In the objective section, it was mentioned that a PXI platform was chosen to meet these end user specifications. PXI platforms are created by NI and provide the modular instrument platform needed for the completion of this project [10]. The PXI chosen for the prototype was a NI PXIe-1073, a 5-slot chassis that can hold a single NI PXI-2800 SwitchBlock. The PXI is sufficient for the proof of concept and can be expanded to accommodate future demands.

Simulation with NI MAX was done with PXI-1036, which is similar to the NI PXIe-1073. As seen in Figure III.1, the NI PXI-2800 SwitchBlock occupies 4 of the slots available, with the added default PXI-8170 embedded controller. The embedded controller, the most common architecture available, dictates processing speed, streaming to disk, etc. [10]. The SwitchBlock is filled with 3 NI 2834 cards which have 34 slots each. In turn, this simulated configuration has a functioning 8x102 matrix, which means that there is room for up to 8 simultaneous connections and 102 different devices.

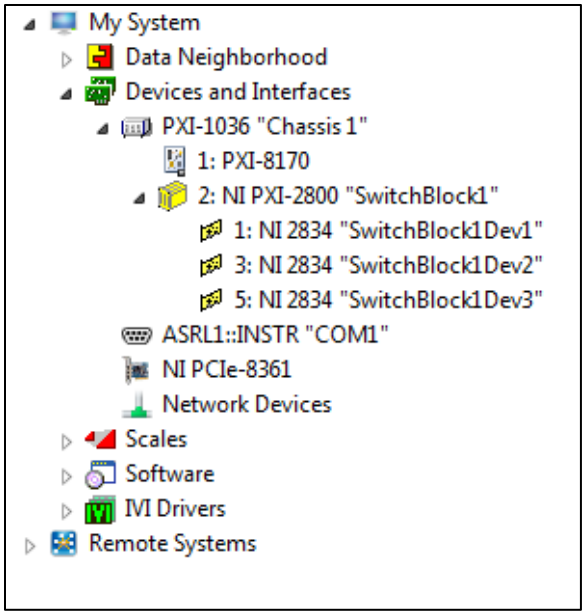


Figure III.1. NI MAX PXI Simulation

Additionally, in NI MAX, the user can open the specific cards and see the state of the switch point relays. Figure III.2 represents the relay positions for the card SwitchBlock1Dev1 from Figure III.l. Some relays have been closed in order to establish a connection across them. In this case, the relay for connecting bus 0 across cards is tripped, b0, as well as the relays for devices 0 and 6, which are connected via bus 0, to relays c0 and c6.

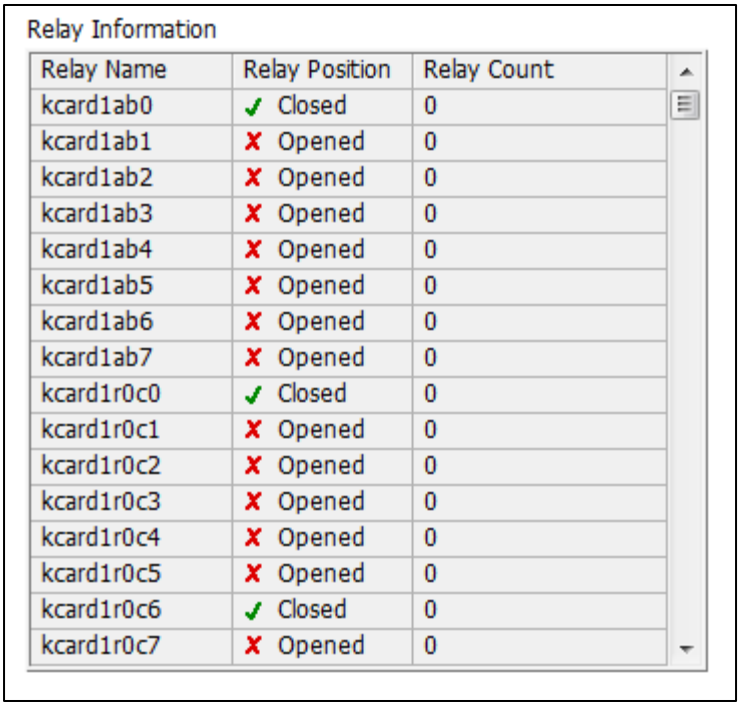


Figure III.2. Relay Positions in NI MAX

A more helpful tool for visualizing the grid of the switch matrix, according to which connections are made in the NI MAX simulation, can be seen on Figure III.3. This representation shows a graphical grid view of the SwitchBlock1Dev1 on Figure III.l. As seen on the figure, there are 4 buses that are utilized: 0, 1, 2, and 7. Buses 0 and 7 are connected to the next card in line, SwitchBlock1Dev2, which allows for more devices to be connected on the already used bus. Some of the devices have been connected as well: c0 to c13 via bus 2 and c7 to c0 via bus 1, etc.

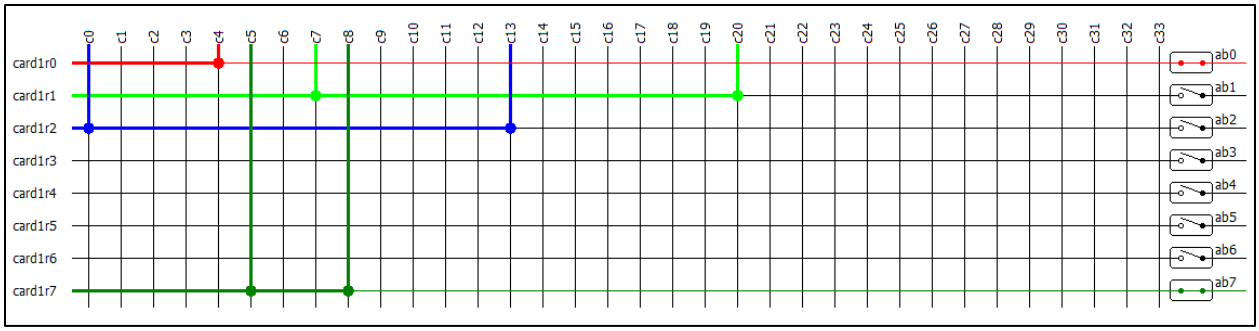


Figure III.3. NI 2834 Card Test Panel

The NI MAX simulation demonstrated that the correct hardware was chosen for the application. The next steps in the development of PrISM was to develop and apply software to control the simulated hardware.

IV Software Simulation

Following the successful hardware simulation, the development of the software began by using the simulated hardware as a test bed for the application. An advantage of using the same developer for the software and hardware is their compatibility with other NI services. NI MAX allows for the simulation of the software in conjunction with the simulated hardware. Therefore, the programming could also be tested with NI MAX before any hardware was purchased.

The modular integration aspect of this project also extends into the software section. The design of the PXI modular instrument was done in conjunction with LabVIEW, the NI software used for controlling NI instrumentation. Therefore, all the programming done in LabVIEW should be able to accept additional functionalities by following a modular approach in the algorithm development.

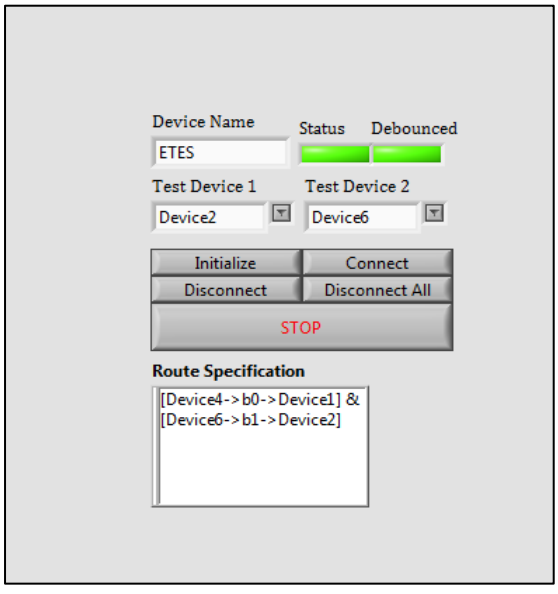


Figure IV.1. Simulated Front Panel

Figure IV.1 shows the front panel of the switch matrix. The user first inserts the device name for the hardware that is used. In this example, the default name given in NI MAX is ETES. The user then selects the devices that will be connected and clicks on the “Connect” button. The connections made show up on the bottom part of the front panel. Figure IV.2 shows device 4 has been connected to device 1 via bus 0, and device 6 has been connected to device 2 via bus 1. Any of these connections can be disconnected by clicking on the “Disconnect” button. The “Disconnect All” button clears every connection that has been made. This setup was used as a proof of concept in order to proceed with purchasing hardware for the PrISM prototype.

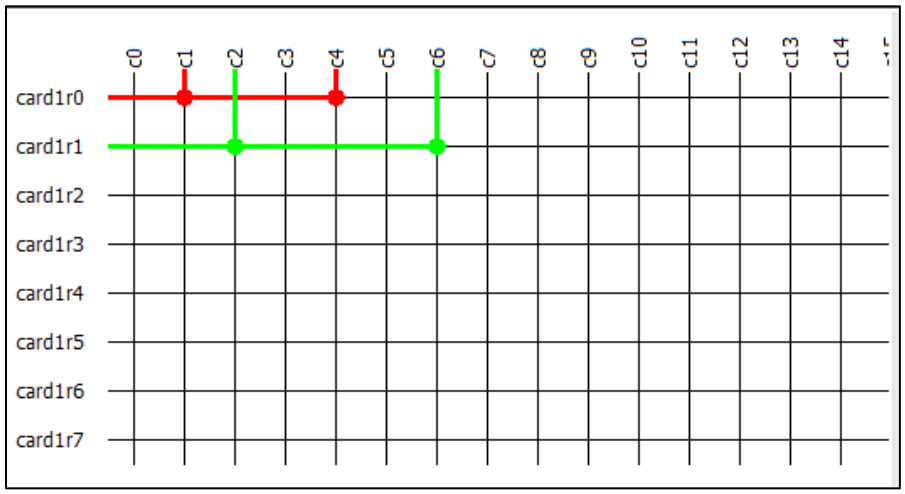


Figure IV.2

The diagram shown on Figure IV.2 is the display that can be seen in NI MAX in the Test Panels tab. This tab shows the state of the relays in the simulated device, in a list where every relay has a name and an open/close state, or in a diagram like Figure III.2, where every pathway is drawn in a different color depending on the bus line that is occupied. As stated previously and seen in

Figure IV.2, device 4 is shown to be connected to device 1 via bus 0, and device 6 is connected to device 2 via bus 1.

The previous results were the proof of concept demonstrating that the simulated software and hardware were working in concert. The project was then allowed to move on to the next stage of assembling the physical hardware which would be used as a test bed for the development of PrISM.

V Hardware Procurement and Setup

The prototype was designed was made to demonstrate each aspect of PrISM. The hardware requirements were presented in Table II.1.

The PXI chassis acquired for this project was a NI PXIe-1073, a 5 slot PXI model. The NI PXI-2800 SwitchBlock carrier requires 4 spots, so the 1073 PXI chassis was adequate in that regard. The card purchased was a NI 2834 8x34 card. These specifications were enough for a proof of concept prototype design, where all of the hardware requirements could be fulfilled. Figure V.1 shows the PXI chassis with the SwitchBlock with one NI 2834 card. The large cable that comes out of the card has the 34 pair pinouts for the copper connections that would be connected to the devices being tested.

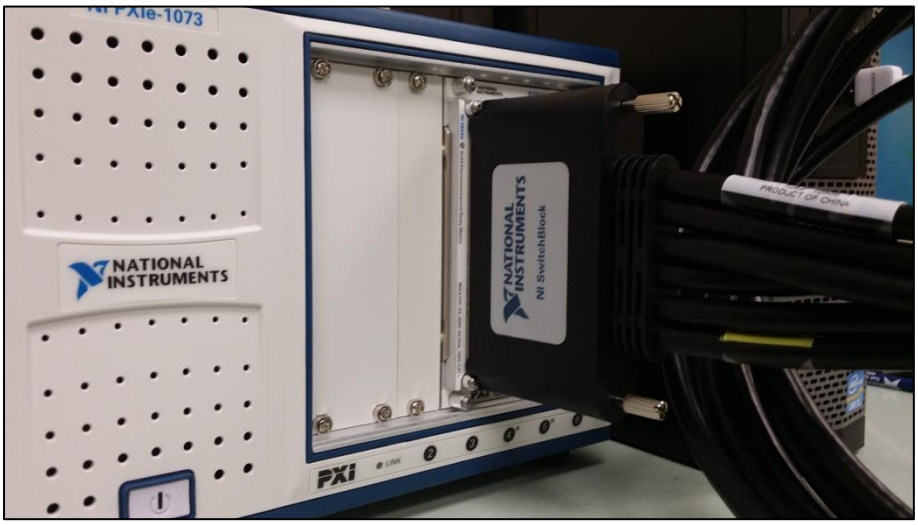


Figure V.1. PXI chassis with populated SwitchBlock

Figure V.2 shows the other end of the cable running from the NI 2834 card in Figure V.1. The copper connectors are connected to a ribbon cable that was used to perform some testing on the device in order to identify the usable ports for the prototype. The green copper connector panel is

where the end-devices would be physically connected to the switch matrix, with 34 of the possible connector pairs leading to a connection made by PrISM.

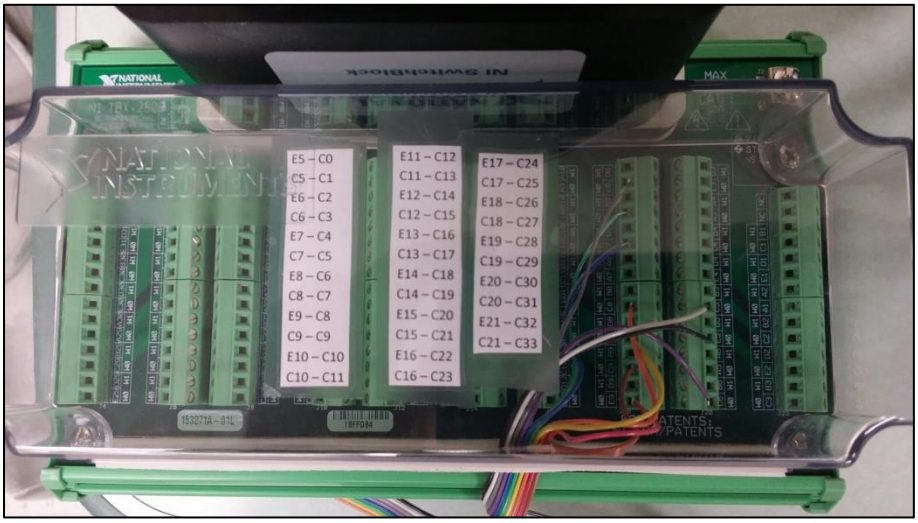


Figure V.2. Copper Connectors to Switch Matrix

VI Algorithm Development

VI.1 Introduction

One of the objectives of the project was to ensure that the device can accept additional integrated functionalities, resulting in the "I" of the name PrISM. With this in mind, the algorithm was developed using an open-ended and template-oriented design. To achieve this functionality, several key points had to be addressed for the success of the project:

- Configuration file(s) for different system parameters
- Program templates for future functionalities expansion
- Features for industry end user

A configuration file is important for the robustness of the algorithm. A large switch matrix, tailored for the specific needs of the end user, requires many parameters to be met. A configuration file containing these parameters would allow the end user to change only the content of the file, without having to make alterations to the main portion of the algorithm. These parameters could be the number of connections available, the names of the devices and their positions, the number of bus lines to allow for simultaneous connections, etc. The configuration files chosen are displayed in Appendix B.

The first configuration file, shown in Figure B.1 in Appendix B, has a description of the format for the user information that the program uses. This allows for the user in charge of the switch matrix to delegate login information to other potential users, and allow different levels of access. As seen from the chart, the first three entries have user names: user1, user2, and user3.

Following a tab delimiter, there are the passwords for these three users, which is “guest” for each, and a designation of 10. The designation of 10 is perceived as a guest user and is allowed limited functionality. The other two users available, “igor” and “jonathan”, have a password of admin, and a designation of 1. The designation of 1 is an administrator, which in addition to the basic functionalities, also has the ability to disconnect or latch onto anyone's connection, as well as disconnect all of the connections at once. These can be adjusted, or more tiers of access can be added, by putting different parameters (besides 1 for administrator and 10 for user) and using them appropriately in the algorithm.

The second configuration file, shown in Figure B.2 in Appendix B, shows the device assignment format that is needed. A five-tiered system was developed for the proof of concept application of PrISM, in which corresponding tiers are:

1) Communication System (Type of group of devices)
2) Rack Name (physical position of the rack in the lab)
3) Rack Number (physical position within the same rack name, usually numerated)
4) Frequency (frequency designations of the devices within the same rack number)
5) Device Name (transmit, receive, record, etc.)

The first tier of communication systems contains different voice switch position communication systems. Placeholder names were used for the prototype version of the device. Similarly, arbitrary rack names and rack numbers were given, as well as commonly used frequencies and transmit (TX) and receive (RX) device names. These can be easily altered by changing the

device information text file. Afterwards, the graphical user interface will recognize the changes as soon as the system is rebooted. Another option for changing the device configuration files was presented via a front panel menu without changing the configuration text files. The current set of tiers and names were only applicable to the proof of concept use of PrISM.

VI.2 Logic Selection

NI LabVIEW works by writing the logic into virtual instruments (VIs) [22]. These VIs are what controls the hardware via LabVIEW. In order to accommodate for the previously mentioned future development of added functionalities, all of the separate processes of this system were designed as separate subVIs. A subVI is a part of the NI hierarchical structure in programming, where a VI can be saved, and called in a different VI as a subVI [23]. This allows for a modular design of a main VI, which can call all of the subVIs that hold different functions as they are needed by the end user. This setup is essential for establishing organization that can accept new functionalities, while having a clear overview from a main VI serving as the staging area of the completed program.

Lastly, as a part of this joint university-industry collaboration, some features were provided to the end user. Having an industry-oriented view of the set of deliverables in an education-oriented project was key to achieving success as shown by other literature [17]. The first of these previously mentioned features is a login system which looks at who is using the system at a specific time, allowing more options available for administrators versus a limited menu for a regular user. This allows an administrator to purge connections which a user might have forgotten about in running longer tests, or add new users into the system. Other features involve improving the design of the graphical user interface, making it more user-friendly and aesthetically pleasing for new users. An addition of a tutorial for new users would make the final system more user-friendly.

The model of computation used to meet the modularity and computing needs was chosen to be a finite state machine. State machines have been successfully employed in designing virtual machines with a need to dynamically respond to a changing environment [24] and in designing faster and more responsive systems than are usually available from vendors [25]. A state machine programmed in LabVIEW has the capability to run continuously and pick and choose which functions to perform at different times, as dictated by the needs of the user. LabVIEW has intrinsically developed queue operations, which allows for the development of a queued state machine. In a queued state machine, all the tasks that are to be accomplished are put in a queue by the producer, and as processing allows it, are performed by a designated consumer. The diagram of a basic queued state machine in LabVIEW is presented in Figure VI.1.

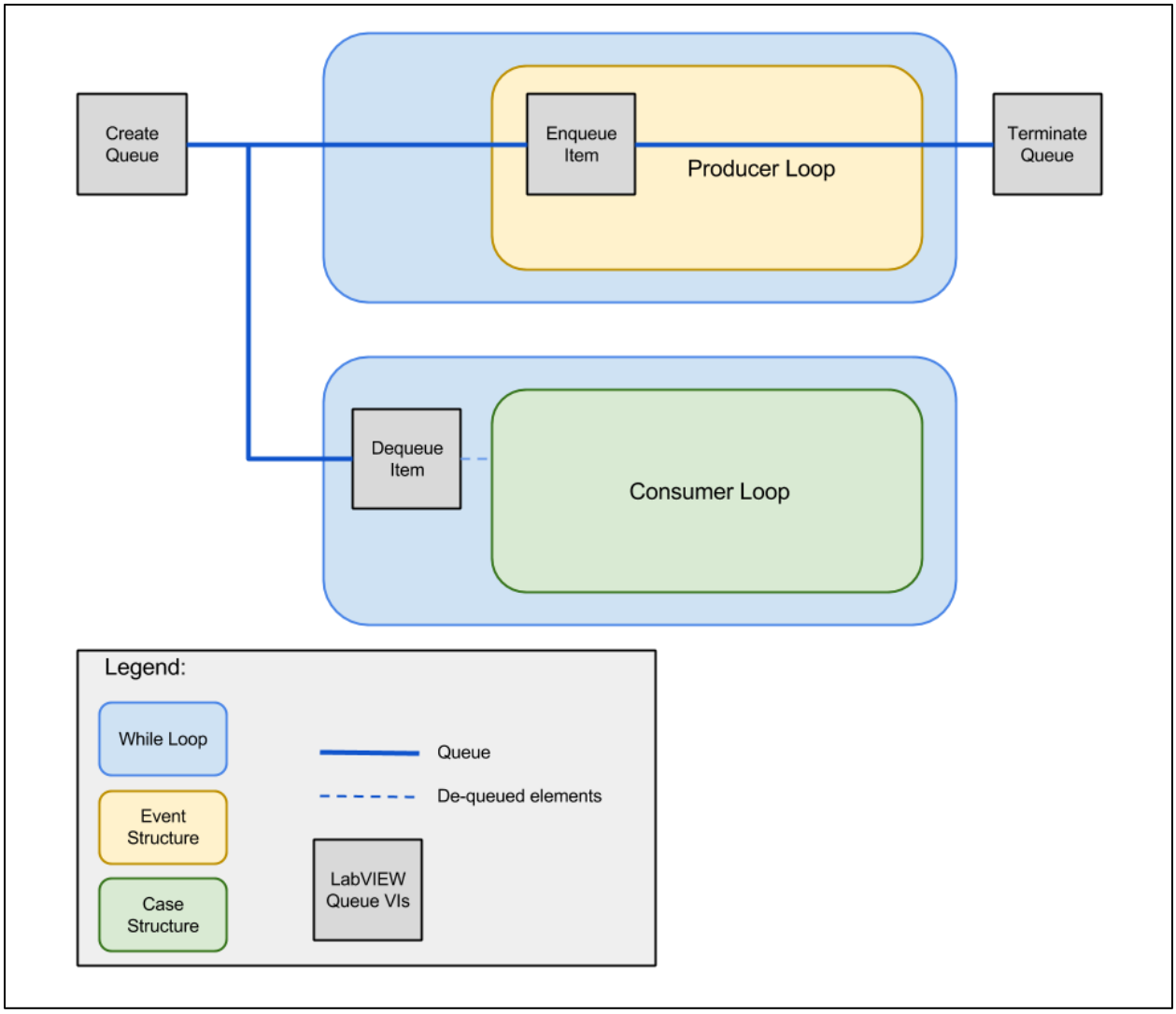


Figure VI.1. Basic Queued State Machine in LabVIEW

LabVIEW has pre-existing intrinsic queue functionality, which allows for the usage of available VIs to perform the queue operations. The producer loop, as depicted in Figure VI.1, is an event structure inside of a while loop. The while loop is active for the entire time the application is up, or until there is an unexpected error which will terminate the queue, display a message and promptly close the application. The producer loop is triggered by different events, each event adding a queue item via the enqueue VI provided by LabVIEW. These events can be user button

clicks, errors, and timers. All of these elements are de-queued in the consumer loop. The consumer loop is a case structure nested in a while loop. The task that was queued by the producer is de-queued in the while loop of the consumer and executed. The case structure of the consumer has a number of different cases that depend on the tasks given by the producer. This ensures that every task given to the algorithm will be completed on a first come first served basis. While this is the functionality of a basic queued state machine, a modified version needed to be designed to suit the needs of this project.

To accommodate the possible future applications that the end user might need, the state machine was altered to accept any number of VIs that can accomplish different functions. As previously mentioned, these functions can be a part of the current setup (connecting different devices, login logic, help items, etc.), or future developments that might be a part of the project (signal generation, signal measurement, remote capabilities, etc.). These VIs were placed in a sequence structure at the bottom of the algorithm, as pictured in Figure VI.2.

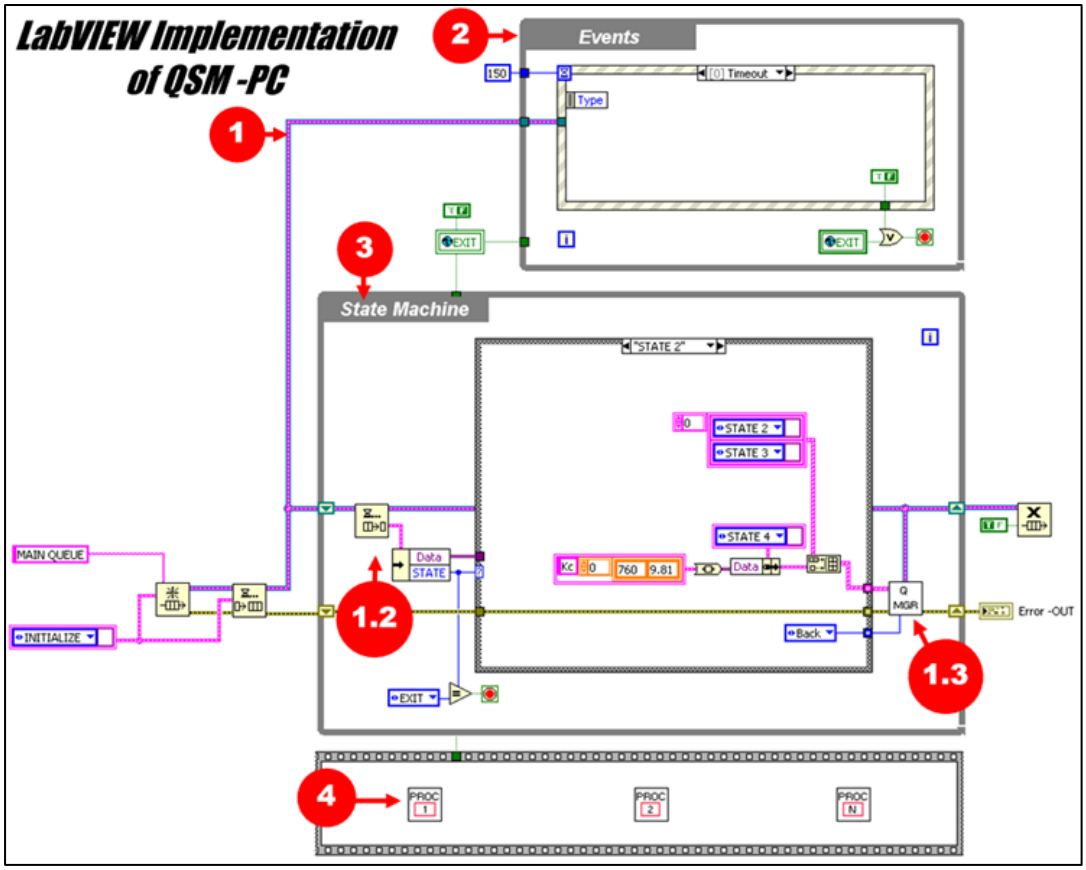


Figure VI.2. Queued State Machine example [24]

The differences in the more sophisticated structure used in the project are marked 1.3 and 4 in Figure VI.2. 1.3 is a queue manager subVI that has the function of changing the order of tasks performed. Some tasks might require the ability to be put ahead of the pending tasks. An example for these emergencies might be an error or hitting the stop button. When either one of those is selected, the preference is for the program to stop. In those cases, a ring control was preselected with 'Front', which will mean that the case is automatically added to the front of the queue. The non-emergency cases, which would contain most of the normal functions required by the program, were preselected with the ring setting 'Back'. This means that these tasks will be performed on a first come, first served basis, as most of those are in line with the normal function

of the program, and don't need priority. The structure marked as 4 is a flat sequence structure. A flat sequence structure in LabVIEW executes whatever is placed inside of it left to right, pixel by pixel. In this case any VIs that can perform multiple functions run in parallel, starting from left to right. The design of these VIs requires them to be running on idle the entire time until being called for to perform some function. In that case, any user input will distribute different states to the parallel processes needed. These VIs that are dumped in the sequence structure will move away from the idle state. This can work for as many parallel processes as the final product might need. Therefore, PrISM can accommodate any future developments in the project by simply adding more Vis in this sequence structure.

VI.3 Algorithm Flow Design

As previously mentioned, for the scope of this stage of the project, there were two main functions that had to be accomplished: connecting and disconnecting devices using a switch matrix, and providing a set of features for the industry end user. The following section will describe the logic in flow charts that were programmed in LabVIEW in order to operate the hardware.

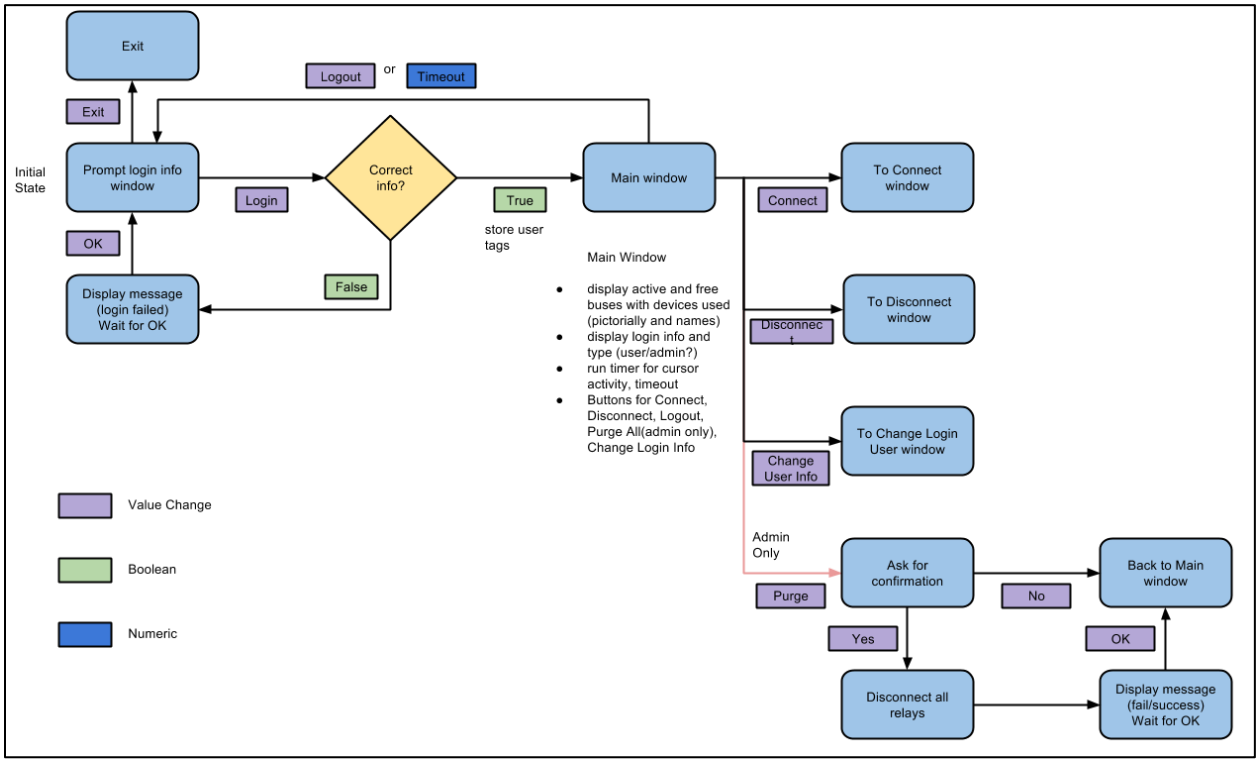


Figure VI.3. Main program flow chart

The main program has the modified queued state machine setup pictured in Figure VI.3. There are some guidelines for appropriately reading the flow charts pictured VI.3-VI.6. Every rectangular bubble represents an action that is performed by the program. Every diamond shape is a query that the algorithm must go through to advance to the next stage. As specified in the

legend in Figure VI.3, the different colors for the parameters represent different variable types. Purple is a value change on a button (e.g. clicking the OK or Cancel button in a simple dialog, or a choice between Connect, Disconnect, Change User Info, or Purge), green is a true or false boolean, and blue is a numeric value. Prior to the main window being displayed, there is a login menu that the user has to navigate in order to access the main window. Depending on the credentials of the user, there are different options available on the main window. The main window itself is the gateway for all the possible parallel processes that can be called by the user at any time. These four parallel processes are Connect, Disconnect, Change User Info, and Purge. As noted in Figure VI.3, Purge is only available if the user has logged in as an administrator. The idle state that the main program is ON, that is: when none of the buttons are being pressed, displays the main window which has the four previously mentioned parallel processes ready for the user to select. In addition to these user initiated parallel processes, there are other events that can trigger parallel processes which not available to the user. These events are safeties like error shutdowns, stopping the application through an emergency button (the close button on the standard window), or other functions available to every other user, such as a logout button or an administrator-generated report**.**

On one hand, the error and exit application functions that are possible events have a high priority level; the queue manager 1.3 pictured in Figure VI.2 assigns them to the front of the main function queue, as well as to the subVI queues which will cause them to shut down promptly. On the other hand, triggering the events that are in the normal scope of the program (such as connecting, disconnecting, user info, logins, etc.), causes the queue manager to put these tasks on the back of the queues on their respective subVIs. For example, clicking on the Connect button

would put an Add Connection task to the back of the queue for the subVI Connect. After this is accomplished, the main program goes back to its idle state, while all of the work is performed by the called subVI. The subVIs and their logic are explained in the following paragraphs.

An important step in the logic of the algorithm is the existence of a connection name, which is saved as a number, and a tag which saves some data about each connection made. Each tag has four components: user, lock, bus, and device. The number that stands for connection name saves each connection made as a number, the first connection gets labeled as "1", the second "2", etc. This is done for logging and report-generation purposes. If an administrator wants to look at the history of the connections made, each report will have a report number, as well as connection numbers saved for each session the application is on. The application is designed to run continuously, and only people with the proper authority can shut down the processes that are ongoing in order to reboot the system or clear the connections in case complications arise. The other data fragment that is carried by every connection is the four tags: user, lock, bus, and device. The user tag logs which user has made a certain connection. Only people with the proper credentials can make connections, and only they can terminate their own connections. Therefore, the system needs to know the identity of the person who made each active connection at every time. An exception to this rule is an administrator, who can terminate any connection at any time. The lock tag checks if the user has locked down a connection. If the connection is locked, no one can latch another device onto the same bus line. If the state is unlocked, then additional connections can be made if the users desire to do so. The bus tag checks which bus is occupied by a connection, a tag mostly used internally in the logic of the connection forming, so that the

first available path is found through the switch matrix. The device tag checks which devices have been connected to the grid, making them unavailable for connections if they have been locked out, similarly only used by the algorithm. This will ensure that there are no conflicting cases of a device trying to make a connection exist. All of these tags are used by the connect and disconnect parallel processes.

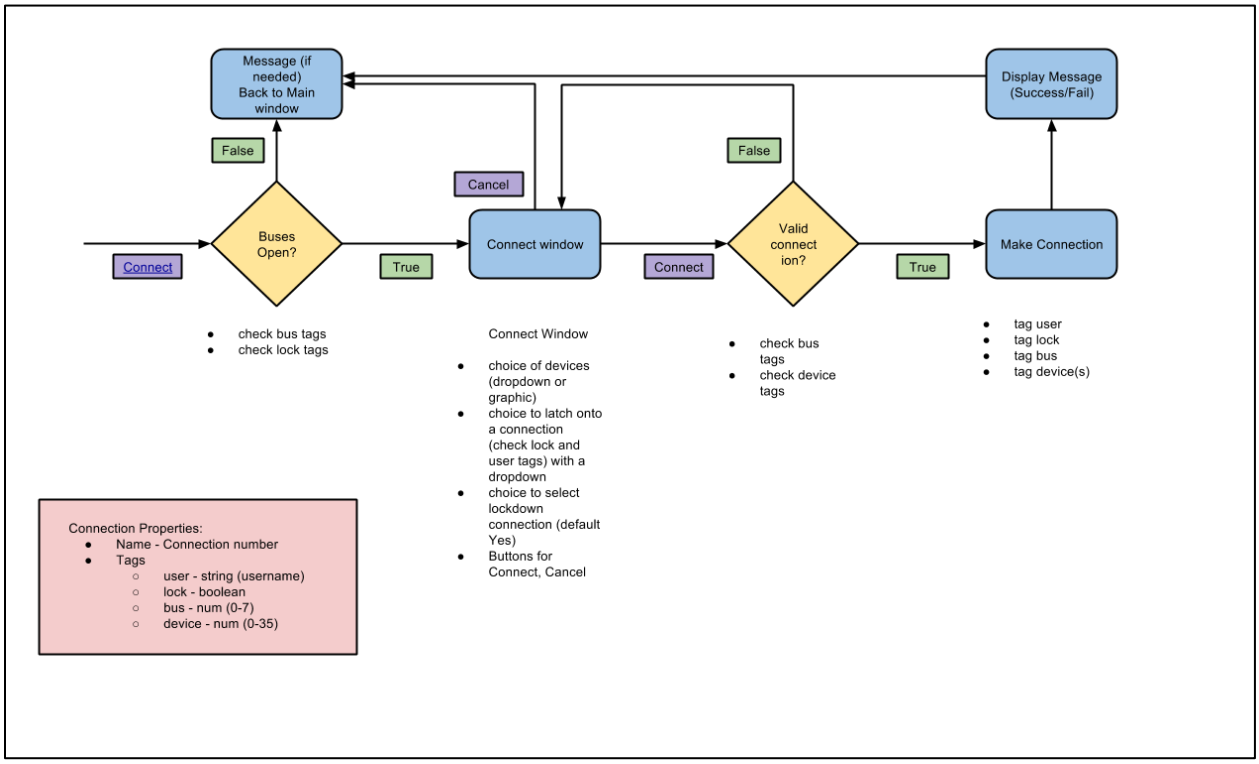


Figure VI.4. Connect flow chart

The flow chart for the Connect subVI is shown in Figure VI.4. The same guidelines apply- the rectangular bubbles are different states and the diamonds are queries for the system. The same data types as Figure VI.3 apply as well. Whenever the user clicks on the Connect button, the system goes into a query where a hardware check is performed. If no buses are available for a connection to be made, the system automatically notifies the user that there are no possibilities

for a new connection to be made. In order to connect a new set of devices together in the matrix, there has to be a bus line available for this new connection to be achieved. In the current hardware connection, up to eight simultaneous connections are allowed. If all eight bus lines are busy at the same time, a message notifies the user that it is attempting to make a connection that the switch matrix is unable to do. After this message, the user is brought back to the main window.

If there are available buses, the system goes to the next step which is the connect window. The connect window has some options for the user to choose. There is a choice of devices to select, a choice to select a lockdown on the channel, and an option to cancel the action and return to the main window. The connect window itself is a dialog pop-up window, and can be removed by selecting connection settings and clicking the connect button, or clicking the cancel button at any time. Both pathways lead the user back to the main menu. When the user selects the devices and clicks on the connect button, the system checks if those devices are available. Once again, there are two possible outcomes: the devices are available and the hardware makes the connection, or one or both devices are not available and the user sees a message and is brought back to the main window.

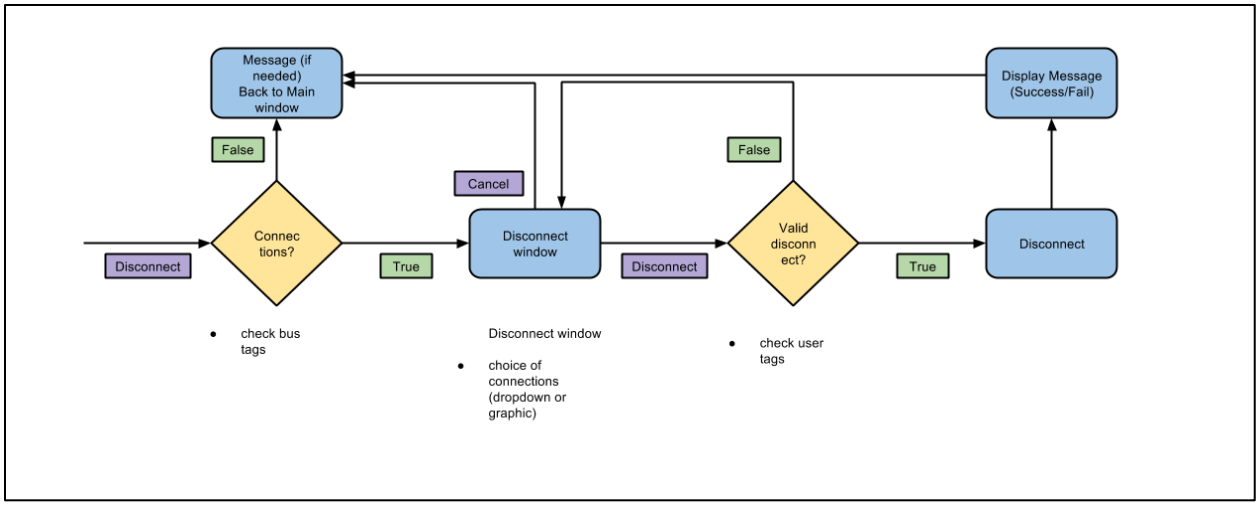


Figure VI.5. Disconnect flow chart

The disconnect parallel process, pictured in Figure VI.5, follows a logic that is similar to the connect logic, with some major differences. When the user clicks on the disconnect button from the main menu, the system checks if there are any connections that have been made by the same user. If there aren't any connections available that the user has the authority to shut down, the system sends a message saying that there are no connections available to disconnect. On the other hand, if there are connections available, the user is taken to a different pop-up window. In this window, unlike the connect pop-up, there is a list of available connections that the user can disconnect. These available connections have buttons that are highlighted, and available to press on. The connections that are unavailable to be disconnected are grayed out, and the user cannot press down on those buttons. From this point, there are two options: the user can either select any of the available buttons and the hardware will disconnect the two devices connected on that bus line, or press cancel in order to return to the main menu without causing any changes to the switch matrix.

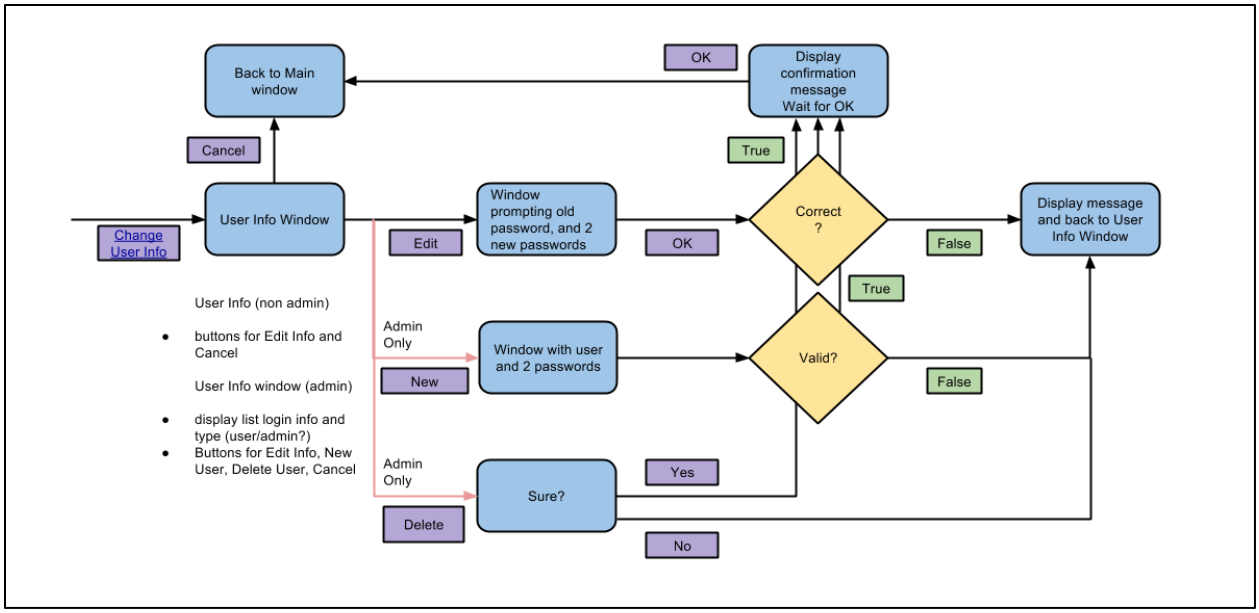


Figure VI.6. Change user info flow chart

The change user info uses a different subVI from the connect and disconnect parallel processes. In this subVI, the user can change their login info, add new users, as well as delete old users. If the user has an administrator login, all of these options are available for selection. If the user does not have an administrator login, only the option to edit his/her own info is available. This was done as a feature to the industry end user, to allow for control over who can change the information of the users. This is important because in order to use the program, every new user needs to log in, which requires clearance from the administrator.

VII Software Results

VII.1 Introduction

The algorithm mentioned in the Algorithm Development section was programmed in LabVIEW, and the resulting product ran in conjunction with the purchased hardware. The function of the program was tested by presenting a series of scenarios in order to test for bugs and ensure that the logic is being executed properly. To make sure that the switch matrix was executing properly, the relays were checked in NI MAX after being tripped by the program. In addition to looking at simulated hardware, NI MAX can be used to observe the function of the connected hardware as well. All of the logic involved in creating new connections and disconnecting the older ones was accomplished. As a disclaimer, all the graphical user interface (GUI) presented in this paper is fully functional and all of the buttons accomplish the tasks they are designed to perform.

VII.2 Front Panel

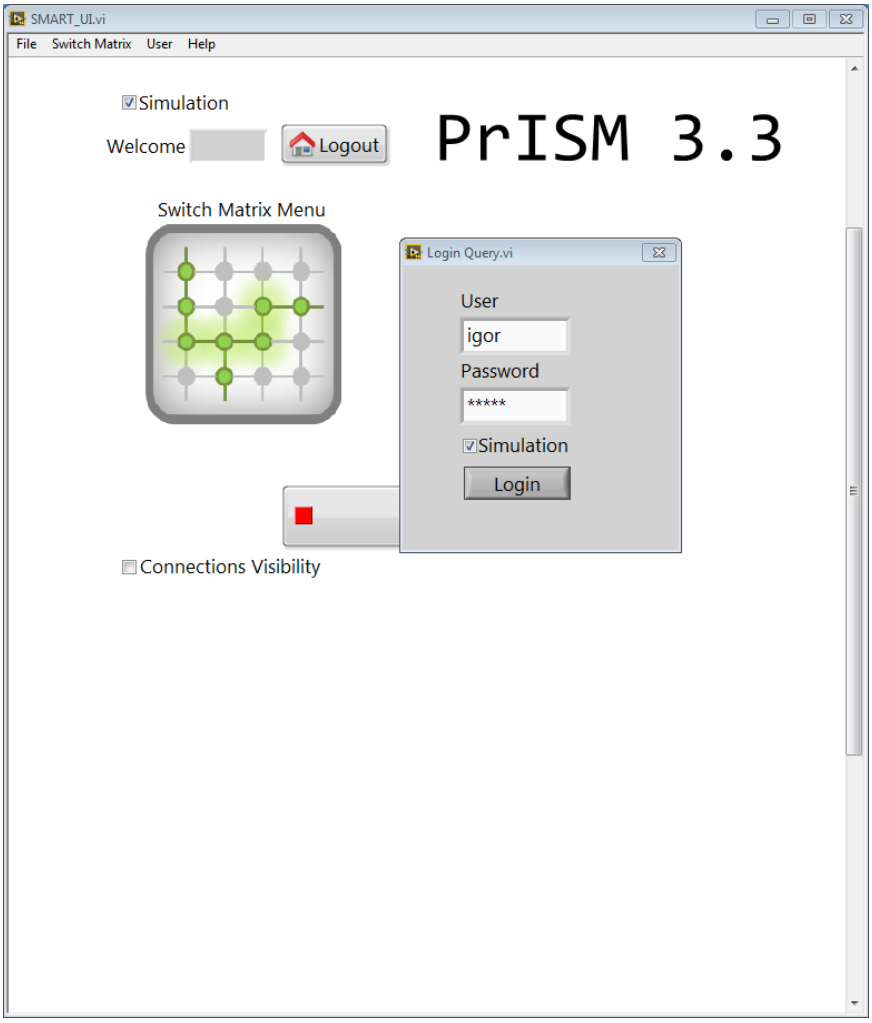


Figure VII.1. GUI Front Panel

The graphical user interface was designed to be as user friendly as possible, while incorporating all of the required functionalities in a simple and clear package. The algorithm was opened in Figure VII.1, showing the default window that appears when the application is pulled up. The user is immediately prompted to enter their username and password, without allowing them to access any functionalities of the software. As seen in Figure VII.1, the pop-up window called

Login Query can’t be minimized before the login is successfully completed. If the user can't type the login information correctly, the program notifies the user which one of the username or password is wrong, and allows the user to reenter their information. The program looks at a configuration file of usernames and passwords, as shown in Figure B.1 in Appendix B. Adding new users can be done by an administrator, and regular users can only alter their own login information, such as changing their username or password. After the login is complete, the user has a multitude of ways to unlock the functionalities of the switch matrix.

Figure VII.2. Switch Matrix Controls

As seen from Figure VII.2, there are multiple ways to access the options available to a switch matrix user. One way is to click on the buttons on the front panel, which have a blue plus symbol for adding, a blue minus symbol for removing, a green plus for latching, and a red minus for disconnecting all devices. The disconnect all devices option, as described in the algorithm development section, is only available for administrators, therefore it is grayed out in this case. Another way to access these software features is through the menu bar. The menu bar has multiple options which can be seen branched out in Figure VII.2. The options are the same: add connection, remove connection, latch connection or disconnect all. Additionally, there are keyboard shortcuts for most of these functions, they are listed on the side of each line. LabVIEW allows for a lot of intrinsic flexibility and functionality to add or remove options from the window.

Each one of the previously mentioned options has full functionality at this stage of the project. The blue plus symbol button is a make a connection button, which will prompt a switch query to occur. The blue minus symbol button is a disconnect button, which will prompt a disconnect check on which connections are eligible. The green plus symbol is a latch connection which functions similarly to the disconnect button. It checks which connections are eligible for a latch. Both of these only show up next to the eligible connections when the logic allows them to. For example, if the user clicks on the disconnect, but none of the connections are theirs or there aren't connections, no button will show up to allow them to disconnect anything. The latch function follows a similar path. If the user who is logged in hasn’t made any, no latch buttons will appear. The system will not allow for the user to latch onto another user’s connection. Also, there is a hard cap on how many devices can be connected at one time, currently set to four. The

disconnect all button is only available when an administrator is logged in, as shown in the current setup.

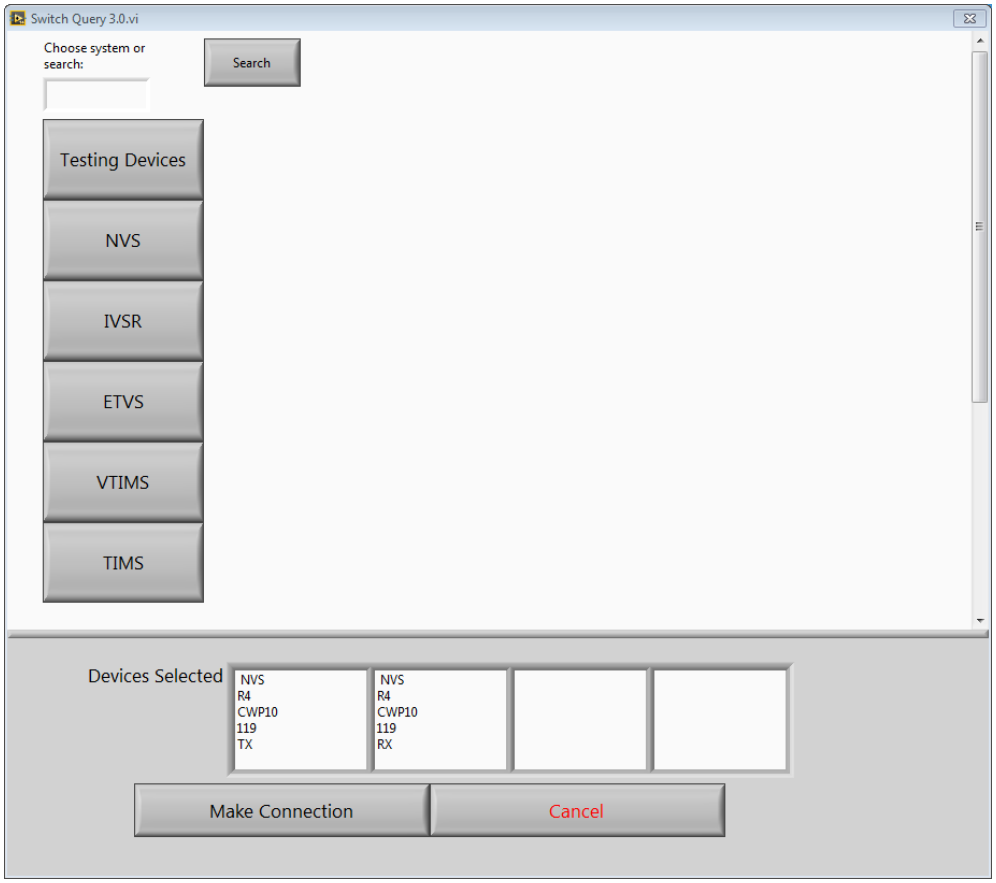


Figure VII.3 Switch Query Menu

After pressing the blue plus symbol make a connection button, there is the switch query that appears as a pop-up window. While the pop-up switch query window is up, the user is unable to go back and click any other buttons on the main window. The switch query allows for a few selections to be made. If the user wants to go back to the main window without making a change, he can click on the cancel button at any time and the switch matrix will remain unaltered. There are two ways the user can add items. With each selection, the system moves through the five

tiers of device groupings to find the device the user was looking for. The five-tiered system was explained in section VI Algorithm Development when showing how the configuration file for devices installed works. Even though the functionality is not quite necessary for the prototype level, a search bar allows for the user to search through each tier in order to find the system, rack name, rack number, or frequency that they need. This would be of immense use when the full switch matrix is deployed, and the user can choose from thousands of devices. Currently the system searches through the few options available with ease. The switch query performs a variety of checks to make sure that the connection can be made:

- Validity of device name
- Different devices need to be selected within the same switch query
- That any of the devices selected are not in use already
- There are enough devices to make a connection (in this case at least two, but no more than four)

To add to the ease of use and the functionality of the switch query, the user can always go back a tier as well, if the wrong button was pressed, in order to save the progress of the other devices selected prior to making the error. This can be a tedious task, especially when there are thousands of devices and small deviations in names are crucial. Going back is achieved by clicking on the previous button which will negate the choice made in the previous tier of device nomenclature.

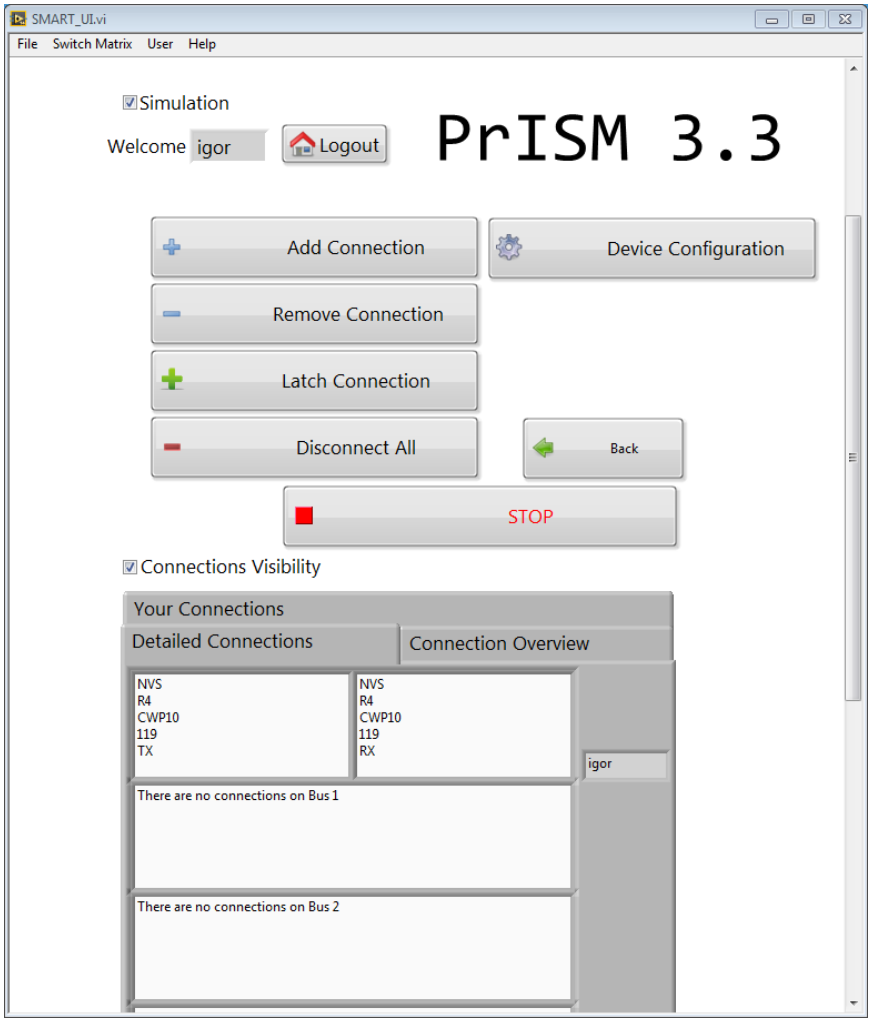


Figure VII.4 Establishing a connection

In this example, following the selection of the devices listed as NVS R4 position CWP10 frequency 119 receive, and NVS R4 position CWP10 frequency 119 transmit from the switch query presented in Figure VII.3, the system will then make the connection via the easiest route it can find. As seen in Figure VII.4, the connection from device 1 to device 3 was made via bus 0, which was the first bus that was not in use. The main window has a large section where the user can see what is being occupied by each of the bus lines. As seen in Figure VII.3, the only connection made was the one that was presented in this paragraph, and all of the other bus lines

are still free. The user can select whether this section is visible by checking the connections visibility box that is under the button selections. This will allow the user to declutter the main window when the connection details are not needed. Additionally, there are other options that allow the user to view their own connections, as well as a graphical overview for all of the connections that have been made.

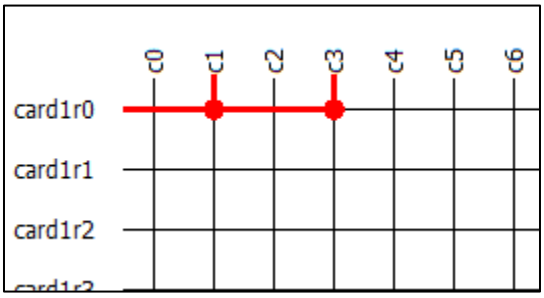


Figure VII.5. Connection made from NI MAX

As seen in Figure VII.5, the connection was made on the physical switch matrix as well. The representation shown in this figure is as shown in NI MAX when viewing the NI 2834 card that is installed in the PXI SwitchBlock. The connection translates what is being shown to the user in the front panel of the GUI to what is happening on the physical switch matrix, where two relays were tripped in order to connect the two devices together. The channels c1 and c3 can be found to house the devices that were shown to the user by comparing the device list found in Appendix B Figure B.2.

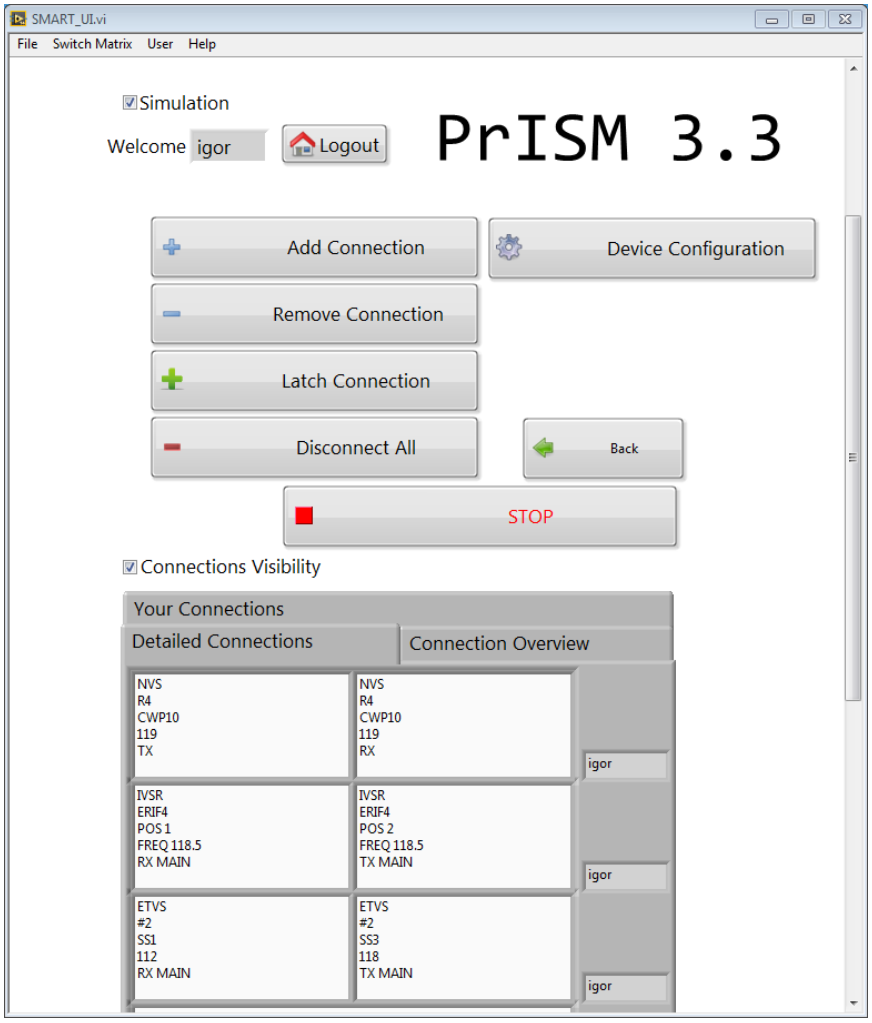


Figure VII.6 Multiple connections

Figure VII.6 shows the result of making more connections on the switch matrix. The program finds the first available bus for each one, and after checking that there are no extraneous connections made, proceeds to make the physical connection. As seen from Figure VII.6, there are three different connections that have been made. Each one of the devices connected is different, since the system doesn't allow for the same device to be connected across bus lines. The limit is four devices per each connection. These can be other transmitters or receivers, or measuring devices. As seen from the tab on the connections visibility section, the user is viewing

the detailed connections window. This allows for the user to see each one of the eight available buses, and each connection shows exactly which devices are connected to it, and which user has made the specific connection. In this case, each connection has been made by the user “igor”.

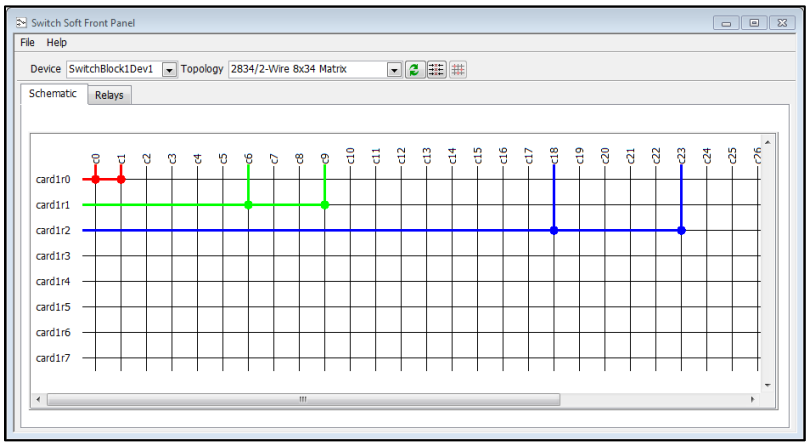


Figure VII.7

Looking at the connections that were made in the main window in Figure VII.6, we can recognize that five simultaneous connections are active. These are supposed to occupy the first five available buses: bus 0 through 4. As seen in Figure VII.7, the NI MAX representation on these five simultaneous connections is exactly as seen in the main window in Figure VII.6. There are five pairs of devices, each connected on a separate bus line from each other, all in the correct spots. Figure VII.6 shows that the algorithm can correctly sort through the information requested by the user, and assign the proper switch matrix relays to be closed.

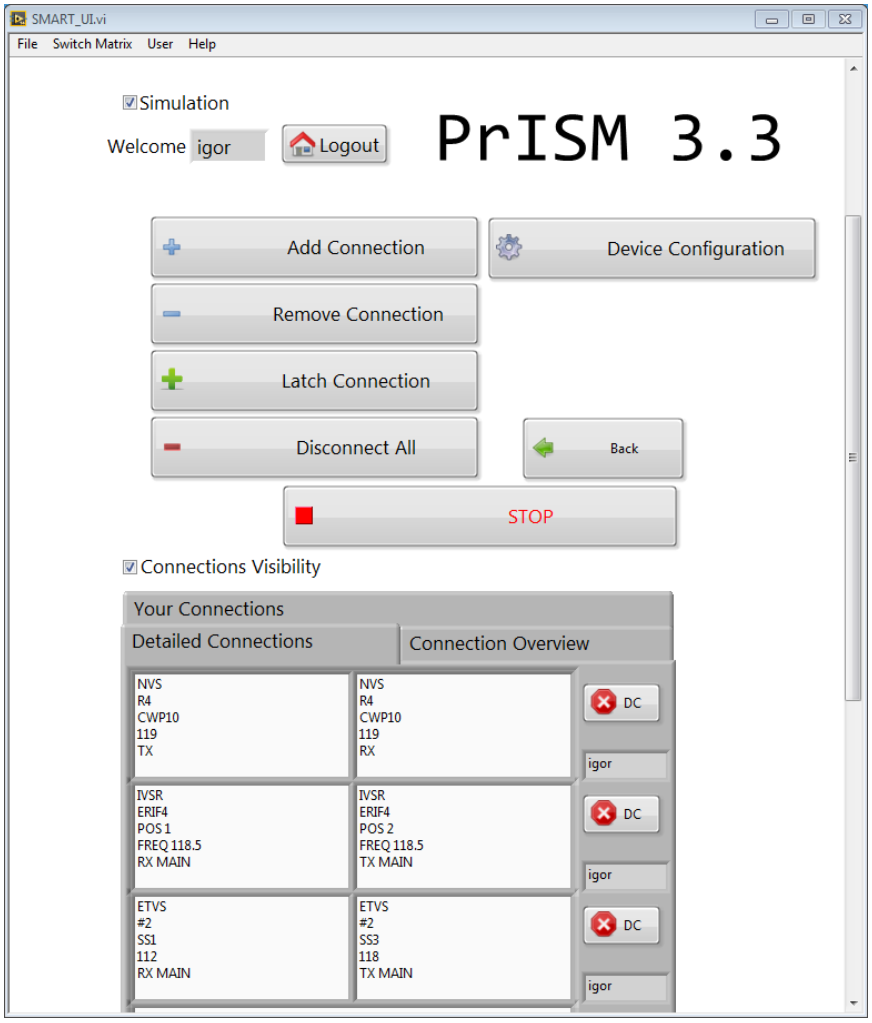


Figure VII.8. Disconnect button population

Clicking on the disconnect button makes disconnect buttons appear next to each eligible connection in the main window, as shown in Figure VII.8. Unlike the connect pop-up, this window will not allow the user to click on any options until he has selected a disconnect option. The algorithm looks at the connections that have been made and populates the buttons for bus lines that can be disconnected. The selection process for the disconnect button population depends on which user selects it. In order to display a button, it needs to run through the following logic: make sure there is a connection set up on the bus, and make sure that the user

that is logged in has made the original connection. This would not allow one user to disconnect a junction that another user has previously created. This is overwritten if the current user is an administrator, who can disconnect any other user's connection.

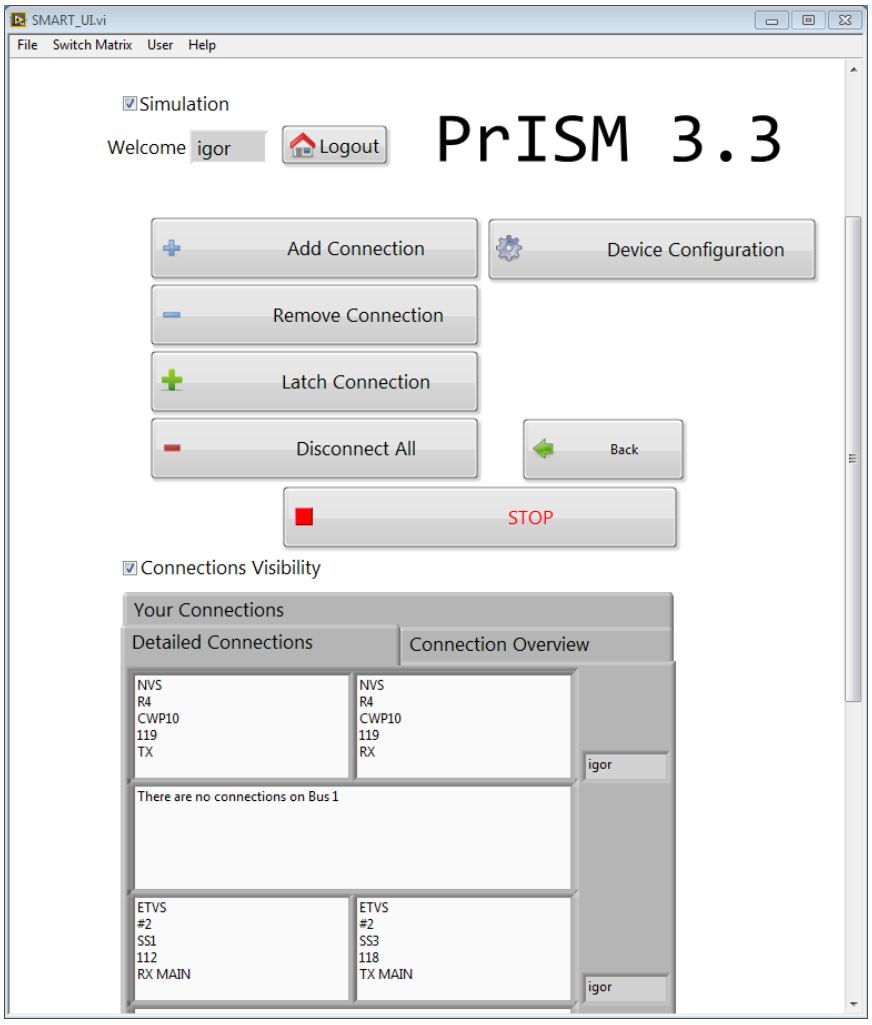


Figure VII.9. Disconnect resulting connection

Figure VII.9 shows the main window after disconnecting the devices that were connected on the second bus line. The program recognized which bus needed to be disconnected, and left the other

connections in place in case there were other tests running on those connections. After this, the devices disconnected become available for future usage.

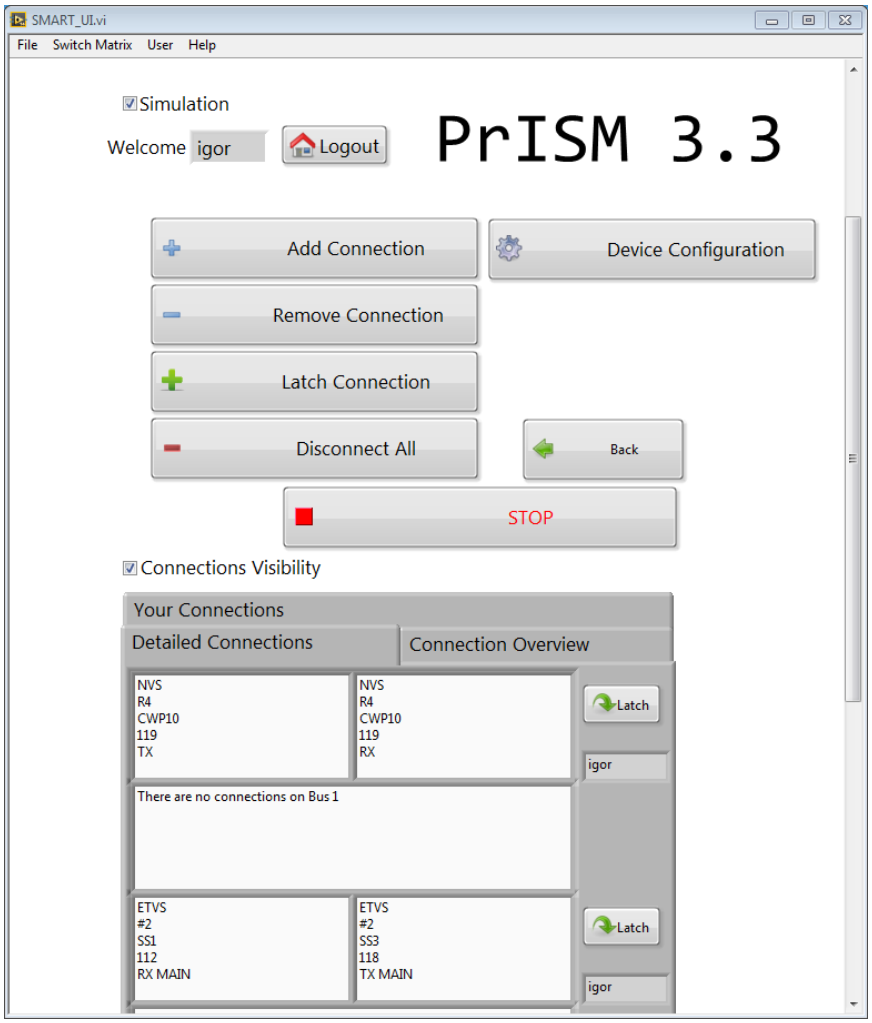


Figure VII.10 Latch button population

Once the user has clicked on the latch connection button, the latch buttons populate the main window next to the connections that are eligible to accept an additional device. This is shown in Figure VII.10, where the latch buttons appear next to three connections. The second bus line

doesn't have a connection, therefore there is no opportunity to latch another device. The first and third bus line have two devices, which means that other devices can be latched.

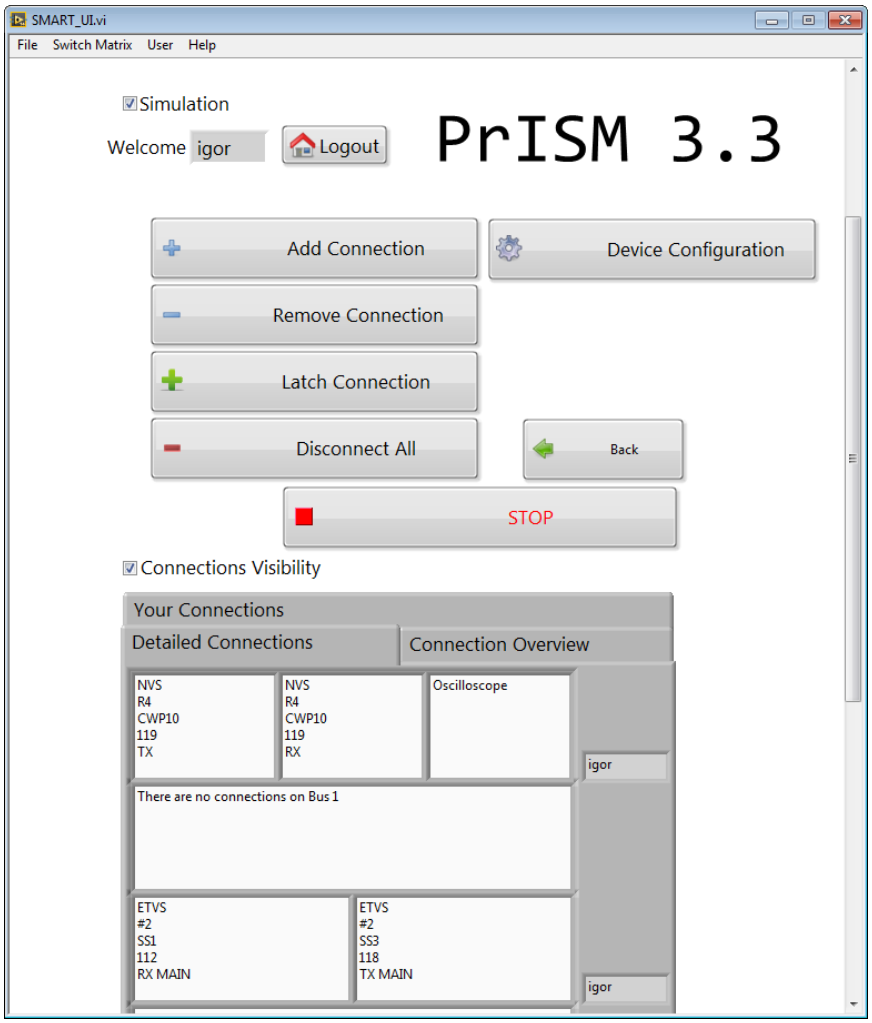


Figure VII.11 Latch outcome

As seen in Figure VII.11, the latching connection was made on the first bus line, where there are three devices connected.

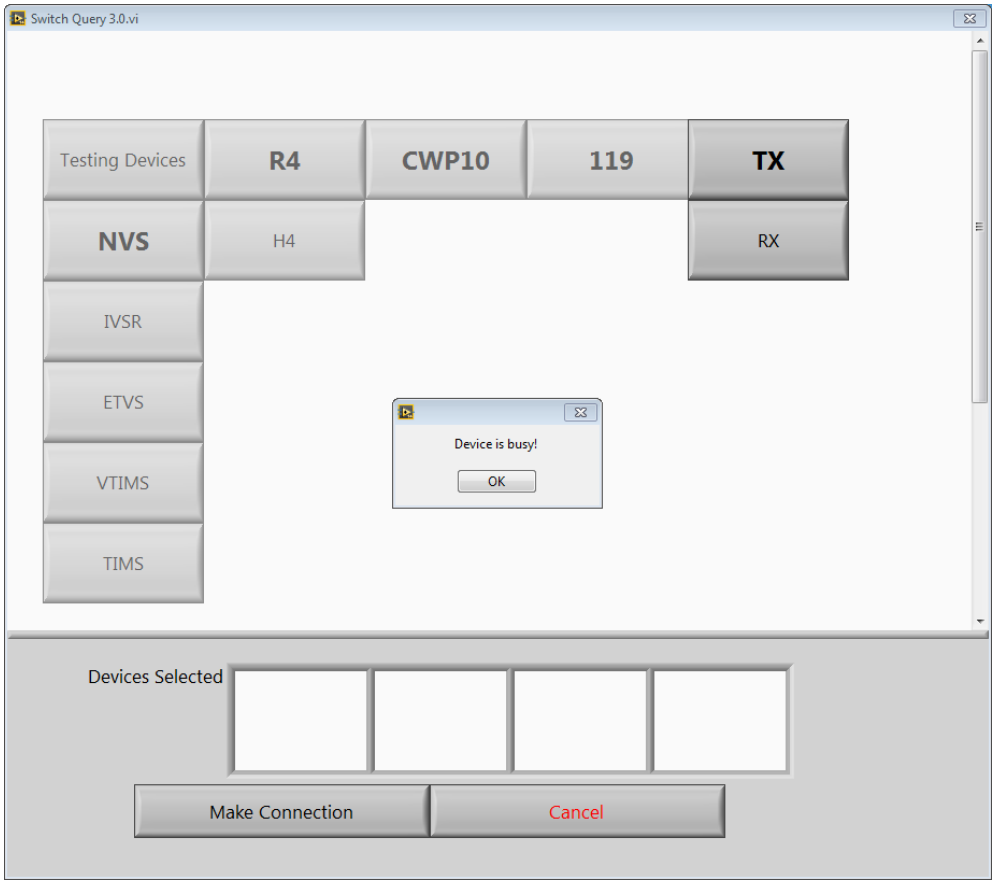


Figure VII.12. Error Messages

There are multiple routes for the algorithm to recognize that there are mistakes in the logic of the user, and send quick dialog windows in order to ensure the user knows why the connection cannot be established. An example is shown in Figure VII.12, in which a dialog notifies the user that the device chosen is busy. This can correspond to a device chosen in the current connection establishment or a previously made connection. If all eight buses are taken, a message will notify the user that another connection cannot be made.

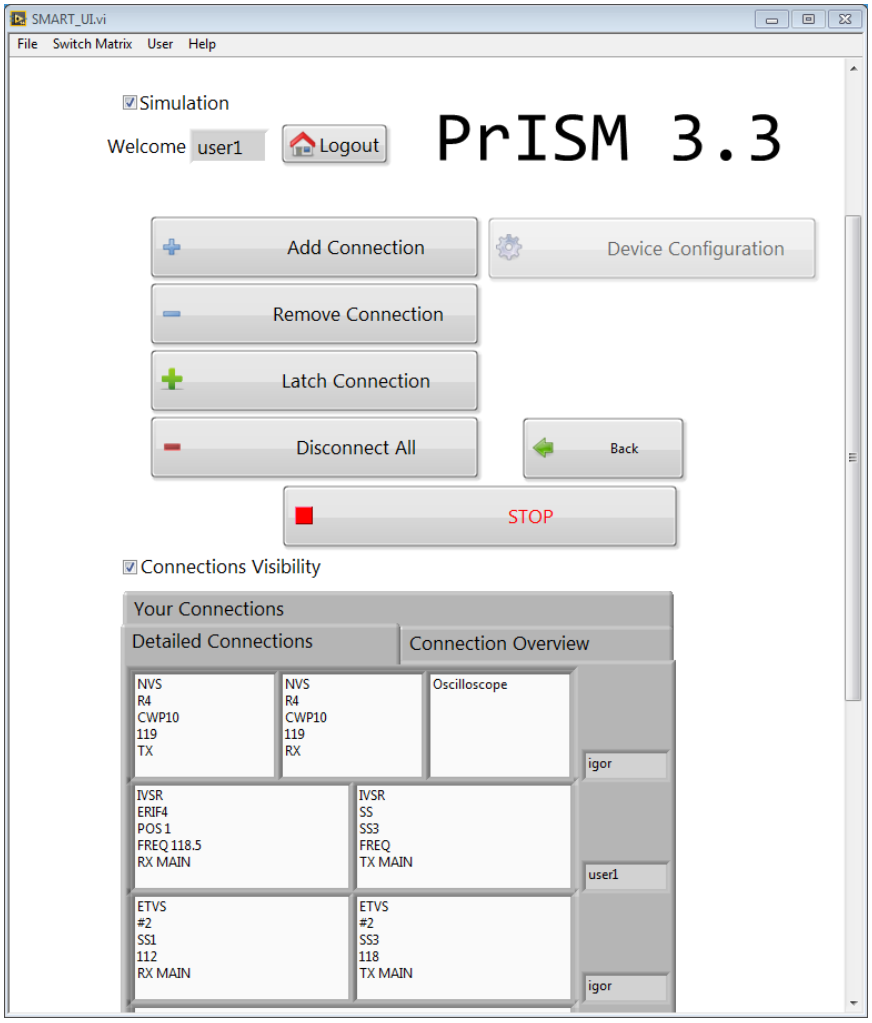


Figure VII.13. Different user login

As seen from Figure VII.13, whenever a different user has logged into PrISM, there are multiple changes that happen to the user interface. The window on top of the screen shows the updated user info. When the new user (user1) makes a connection, it is reflected next to the adjacent bar (in this case, on the second bus).

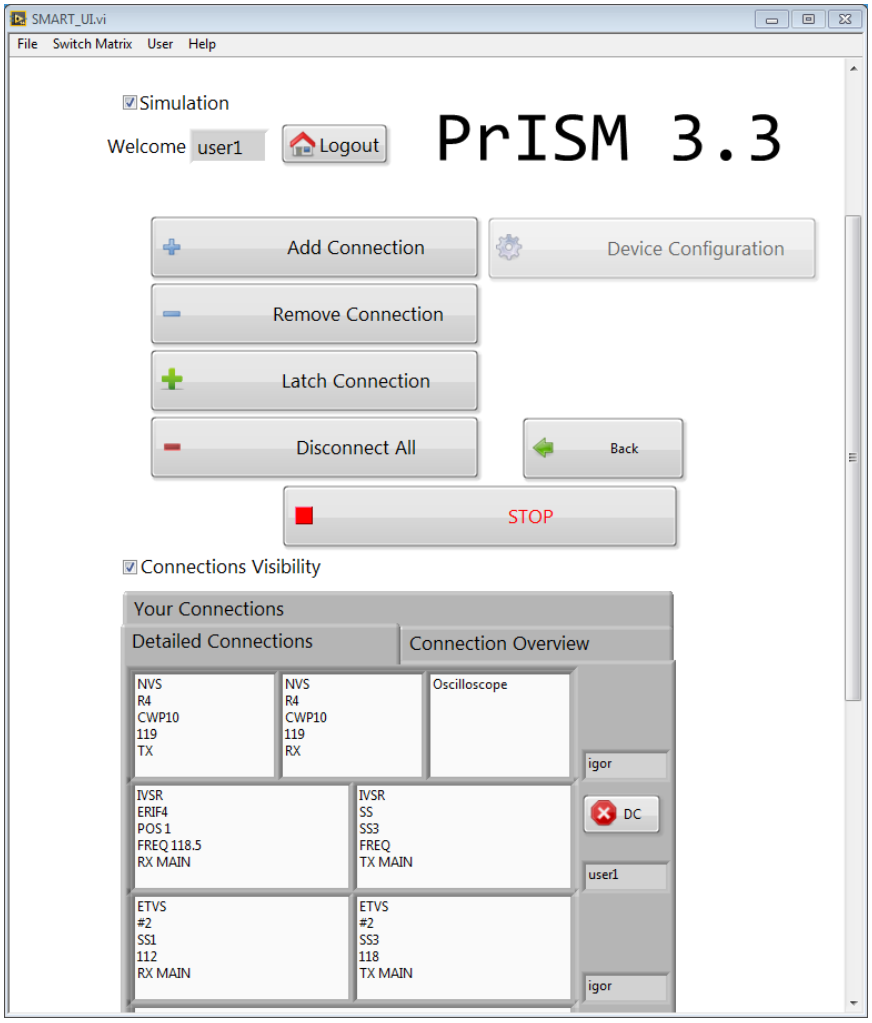


Figure VII.14. Menus for different user

The algorithm doesn't allow for the new user to change any of the already established connections from the previous user. In this case, user2 can only disconnect or latch onto his connection on the second bus, and all the other buses are not allowing the option to be changed in any way.

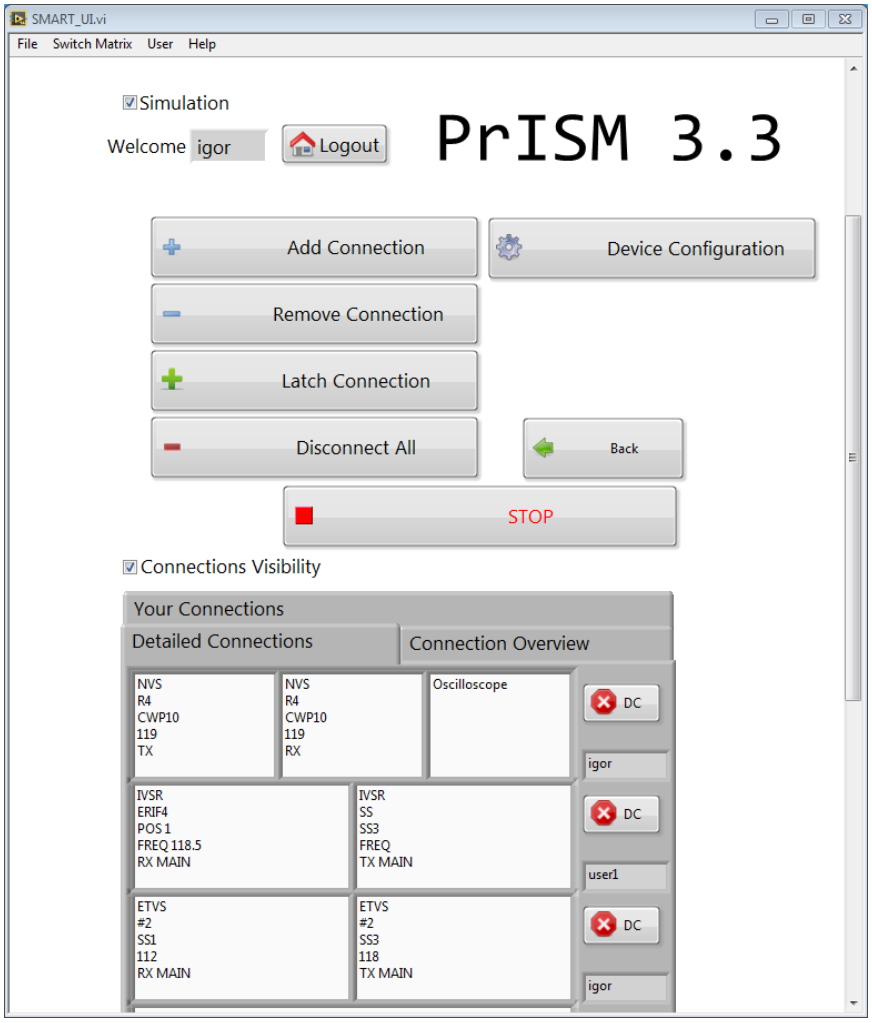


Figure VII.15. Menus for different user (administrator)

Unlike the previous example, if the user is an administrator, the system recognizes this, and allows for any changes to be done to any of the connections. This can be seen on Figure VII.15. Another function available to administrators is the disconnect all button which was grayed out in all the previous screenshots of the front panel. With the addition of this button, the full functionality of the program is available to the administrator.

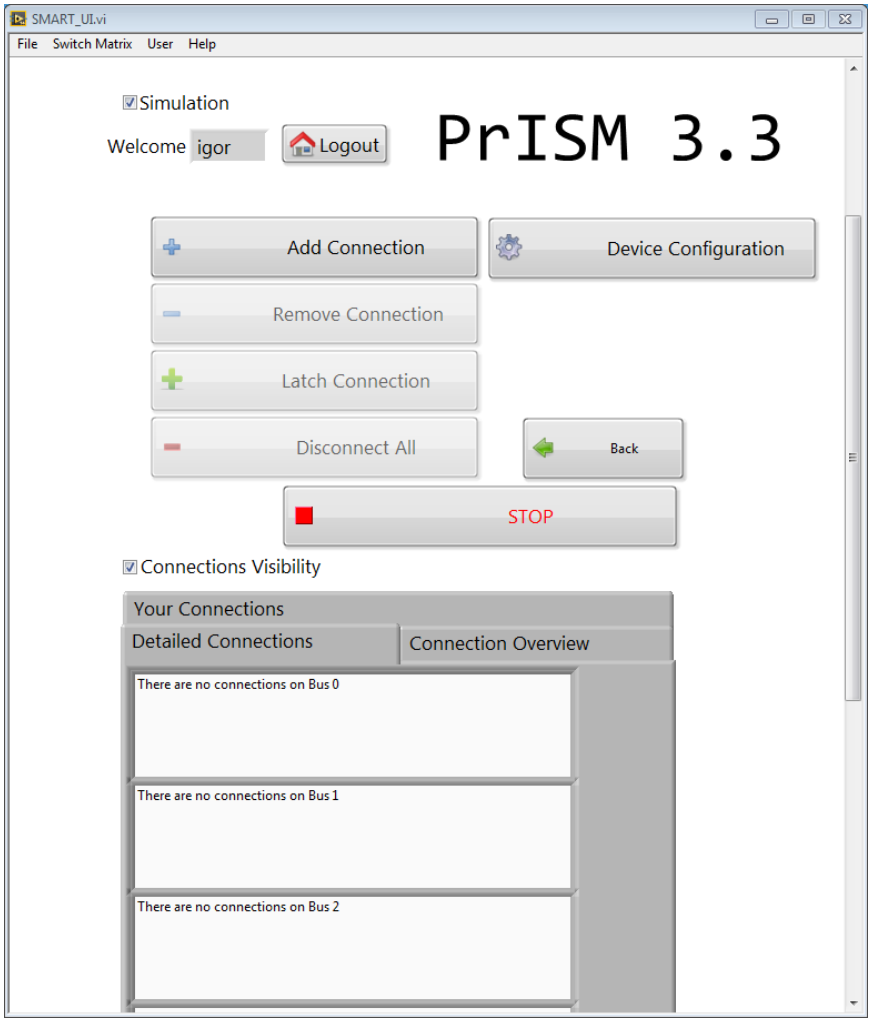


Figure VII.16. Disconnect All result

Figure VII.16 shows what happens after an administrator has clicked on the disconnect all button. All connections have been removed, and all buses are free to be used.

After demonstrating the 3.0 version to the engineers for the industry representative, they made some suggestions about features they would like to see as we were approaching the testing phase of PrISM. These features were the ability to configure the devices available for connections depending on their physical location, as well as the availability of a help menu and

a tutorial. All of the features were made available first in the 3.1 version, and then finally in the debugged 3.2 version of the program. This was the version used for the completion of the first round of the testing section. Version 3.3 was developed with further improvements, and this was used for the second round of tests.

VII.3 Configuration Menu

The configuration devices option was made available only to users with administrator access. This was done in order avoid situations where any user might change the information regardless of their full knowledge of the system. The menu looks slightly different for an administrator in this version, with the addition of one more button when the user has selected the switch matrix. Version 3.2 and subsequent versions also includes enlarged icons for better visibility in demonstrations and presentations. After clicking the button, the window as shown in Figure VII.17 shows up. On the left side, the user is allowed to write the information of the device they want to add, and on the right, there is a list of the devices that were configured at the time. The devices have the same tiered labels, with the addition of the sixth column with the physical positions of the channels for the specific device, and the seventh column with the number of connections for each.

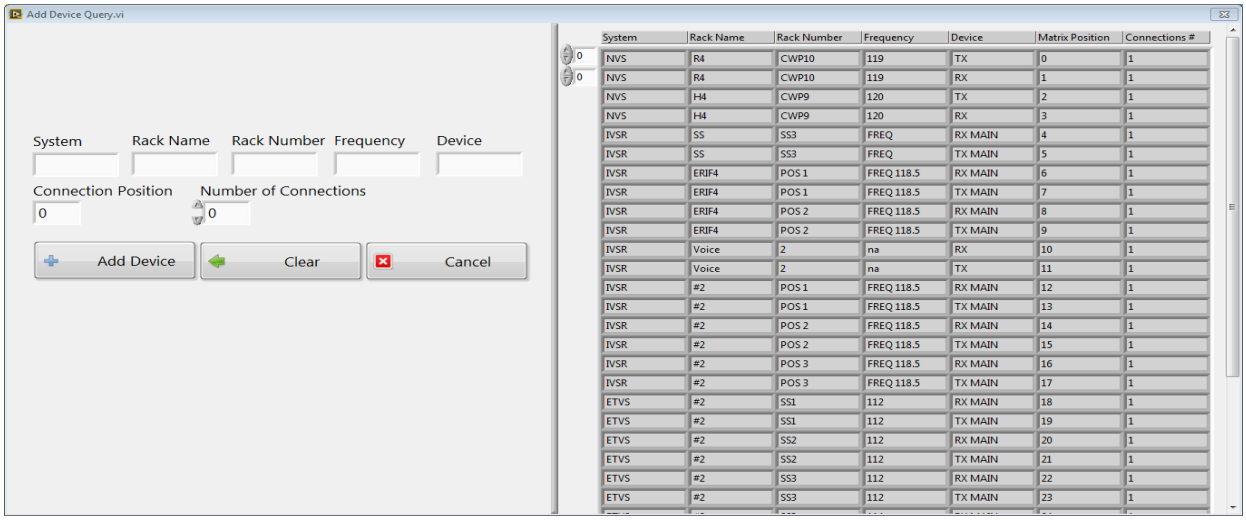


Figure VII.17. Configuration Query display

VII.4 Help Menu

The help menu was added as another desired functionality of the PrISM GUI. Instead of going for a generic glossary help menu, the design involved an interactive help menu, where the users can see screenshots of the GUI and navigate themselves. This is depicted on Figure VII.18, where the main menu is pulled up, with buttons pointing to all of the functionalities that can be reached from the main menu.

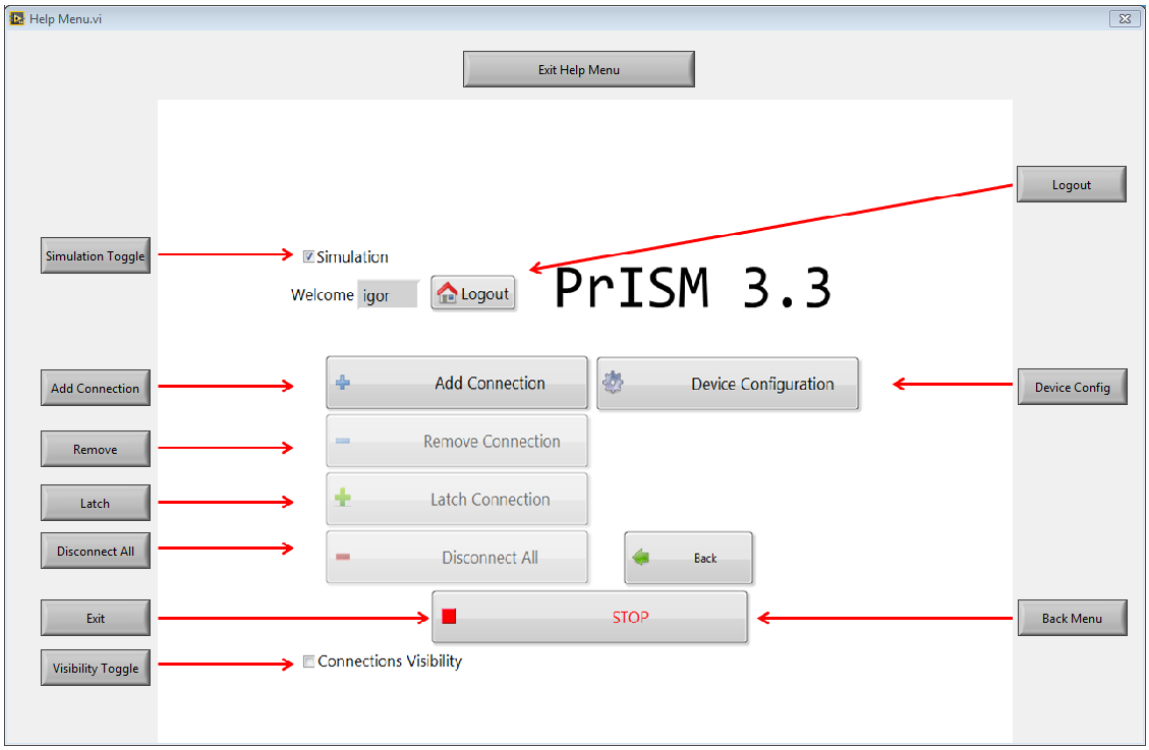


Figure VII.18. Help Menu function

VII.5 Tutorial Menu

The last feature that was included in PrISM 3.2 was the addition of a tutorial option. This would pull up a different window as shown in Figure VII.19. Whenever the user goes through the parts of the tutorial, their progress is tracked. This can be printed out for confirmation that the new user has learned how to operate PrISM. The tutorial is available for new users of the program; completion of it is required the first time they log in. Administrators and returning users can use the functionality of PrISM without completing this tutorial again.

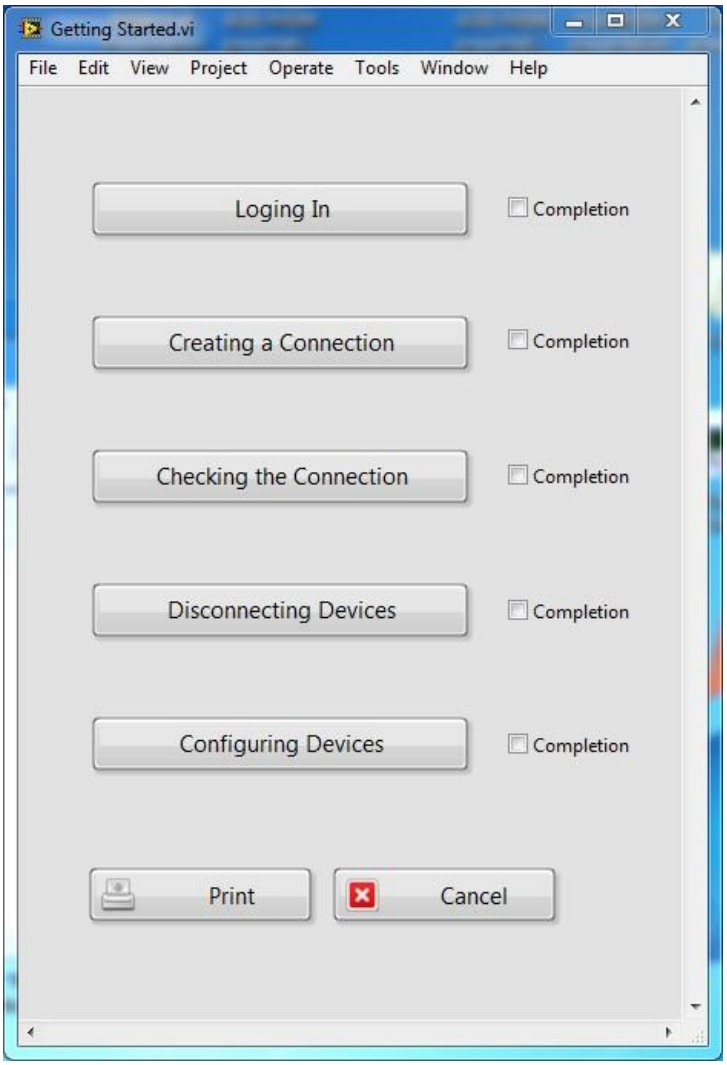


Figure VII.19 Tutorial Menu Function

VIII Results

VIII.1 Introduction

Following the design of the software and hardware setup, PrISM was deployed at the OKCET laboratory to begin making connections between real world communication devices. To test the functionality of PrISM, a series of tests were performed at the laboratory with the cooperation of their engineering team. Verification was performed by connecting different configurations of the current switch matrix, PrISM, communication devices, and testing devices. This was done in order to ensure that every scenario was covered, proving that PrISM did not add any distortion or delay to the signals observed.

VIII.2 PrISM Connections Testing

In order to check for communication device signal quality with the addition of PrISM, the first tests were performed by connecting it to the demarcation point. The first devices connected to the demarcation point were a battery and a digital multimeter. This setup is depicted in Figure VIII.1.

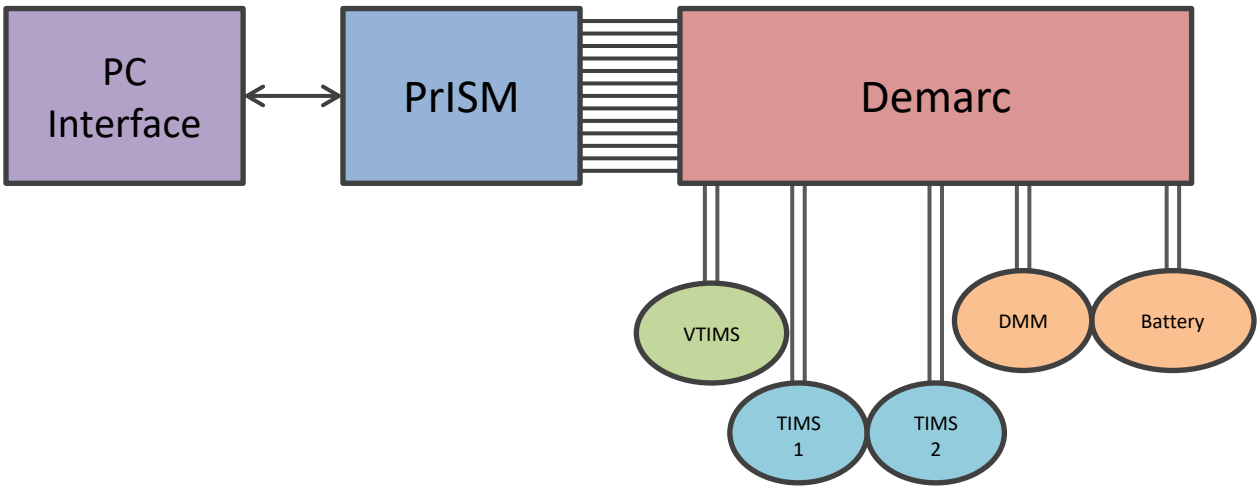


Figure VIII.1. Demarcation point setup (no communication devices)

The following preliminary tests were performed prior to proceeding with the connection to actual communication devices:

- Connection
- Multiple Connections
- Impairment
- Signal to Noise Ratio (SNR)

The connection check was done solely to ensure that a signal was going through the hardware connection points as expected from the graphical user interface. This is depicted in Appendix D.1, where there are two different values for the signal loss across the connection. These two different losses are due to using two different devices. One was using a HALCYON Transmission Impairment Measurement Set (TIMS), shown in Figure VIII.2, and the other using the Virtual TIMS (VTIMS), another graduate student project from UCO. The power loss across the connections was minimal and had statistically insignificant numbers, most likely due to the wire connections in the demarcation point.

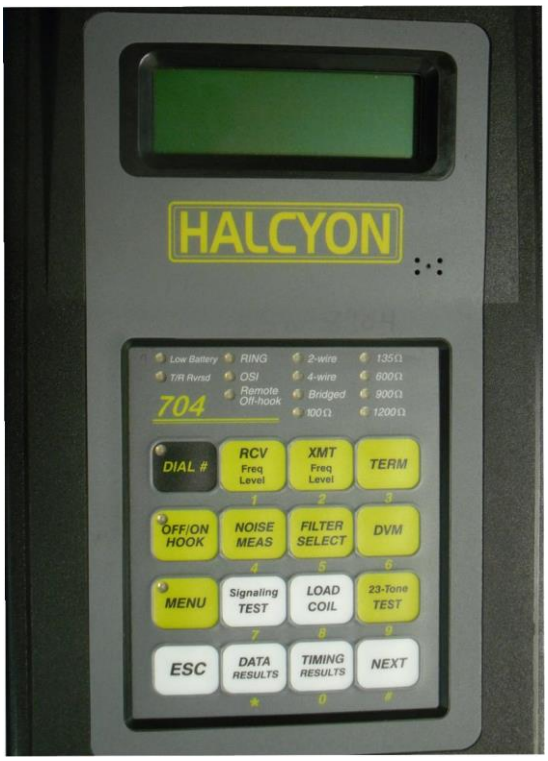


Figure VIII.2. HALCYON TIMS

A test for multiple connections was ran by making sure that multiple devices can be connected at the same time without interference between them. This test connected the devices on multiple channels, and the results can be seen in Appendix D.2.

The second round of tests was run solely with the VTIMS. The tests were a series of impairment tests that can be reviewed as a part of Appendix F. The report automatically fills out failed tests as red. There were no failed tests, hence there were no red fields in the results from the VTIMS. The test results for multiple channels are shown in Appendix D.3.

The last connection test to the demarcation point was signal to noise ratio (SNR). The first test, shown in Appendix D.4, shows the different end devices that were tested. In order to ensure that

there was no error from the measurement devices, the TIMS and VTIMS were used as transmitter and receivers in various combinations. The SNR for all of these configurations was at least 50dBrnc, which is beyond the IEEE communication standard and those referenced by other papers involving research with voice quality [26] [27] [28] [29]. The second SNR test was performed on different channels, as shown in Appendix D.5, to show that there is no noise being introduced by the PrISM hardware. Once again, all the values were beyond the IEEE communication standards.

VIII.3 Communication Devices, VXI, and PrISM Testing

The second round of tests involved connecting PrISM to the communication devices inside the OKCET laboratory, the final demonstration of its full functionality. This was done as demonstrated in Figure VIII.3. PrISM was connected to the devices through the existing switch matrix instead of directly to the demarcation point because these connections were going to be made to a live device. Another set of connections had to be made to avoid disconnecting any live communication devices to the existing switch matrix. Regardless, if the tests through PrISM and the existing switch matrix are passing, then going only through PrISM would only enhance the signal quality. This is because bypassing the current VXI (VME eXtensions for Instrumentation) Switch would only remove hardware for the connection between end-devices. The tests were performed on selected communications devices and testing equipment from the OKCET lab.

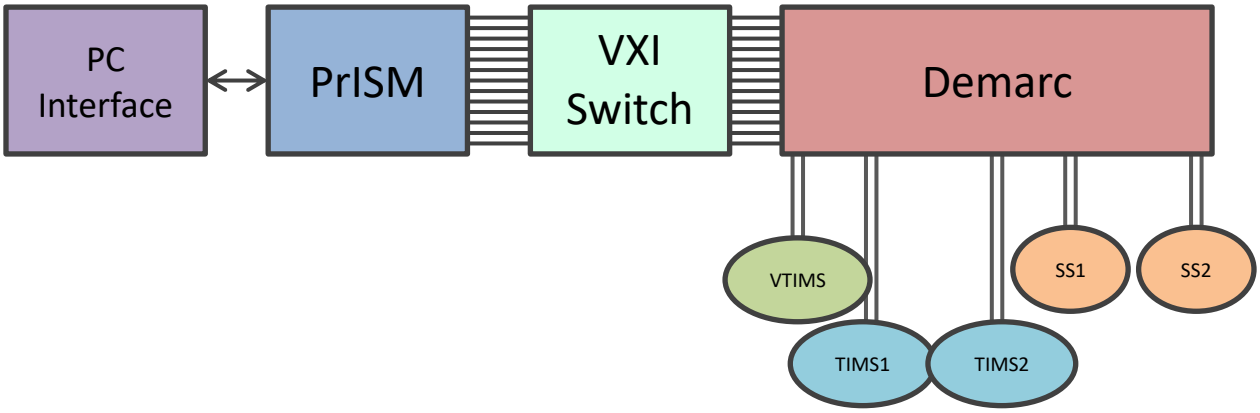


Figure VIII.3. Communication devices setup

The first test was performed in order to make sure that the connections through PrISM and the existing switch matrix were being made as requested from the graphical user interface. The tests ran were visual and audible, checking whether the dial tone and an audible voice signal were

going through. The dial tone was sent from device SS1 and picked up by device SS2, and vice versa. When one would dial the specific input, the other would signal that there is a line that can be picked up. After picking up the line on both sides, a standard voice check was confirmed. The parties on both sides of the line could hear each other loud and clear. Both tests results are shown in Appendix E.1.

Following the basic voice check test, a frequency analysis was set up by using the TIMS and VTIMS as transmitters and receivers in various configurations. Because of the notch filters inside the SS cards, an altered frequency of 704Hz was used in order to test for any anomalies. The results were consistently showing a loss of 16dB with no measurable frequency shift. The 16dB loss is standard for passing through two SS cards. The results of these tests are shown in Appendix E.2.

However, the most important tests were performed in order to check the signal quality that is being transmitted by looking at the voice quality. This is of paramount importance because these devices are designed to transmit human voice between airplanes, radio control towers, etc. Adding any signal impurities by introducing new hardware would make the validity of the replacement system hard to justify. This was done using a Dual Universal Telephony Adapter (Dual UTA), by looking at the results for one-way-delay (OWD), Perceptual Evaluation of Speech Quality (PESQ) [30] [31] [32], and Perceptual Speech Quality Measure Mean Opinion Score (PSQM MOS) [33] [34]. The tests were done in parallel by communication between SS1 and SS2 through PrISM and the existing switch matrix (as shown in Figure VIII.3), as well as by connecting the devices only to the existing switch matrix (skipping PrISM). If the voice quality

by going through PrISM is insignificantly different from the one without PrISM, then we can conclude that the test has been passed. As seen from the results in Appendix E.3, the results for going through PrISM are very similar to the ones without it. This means that PrISM introduces negligible delay or signal quality deterioration. Some of the tests even show an insignificant improvement by connecting PrISM, this being only due to the variance of the tests.

VIII.4 Communcation Devices and PrISM Connection

For the third round of tests, PrISM bypassed another switch matrix, connecting directly to the demarcation point. The full setup of the system, including PrISM in the connection, is shown in Figure VIII.4.

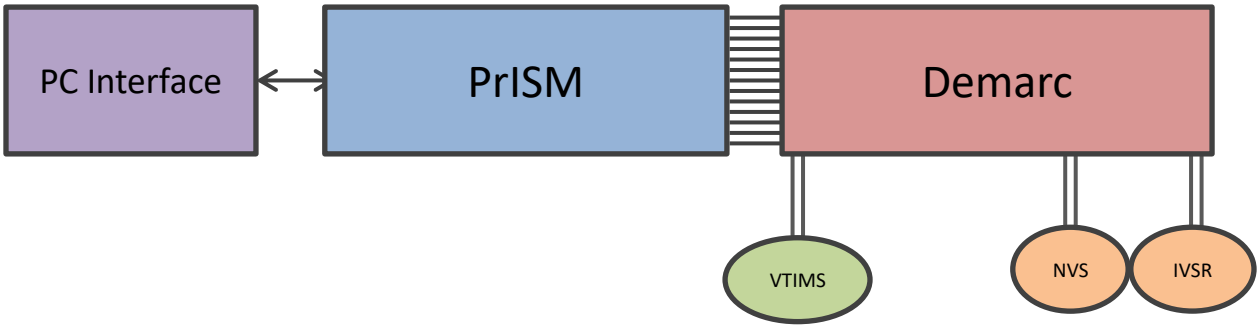


Figure VIII.4. Demarc Only System Testing Overview

The connections of the systems were done using the automated connection making in PrISM. The menu as depicted in PrISM is shown in Figure VIII.5, and the NI-MAX hardware representation is shown in Figure VIII.6.

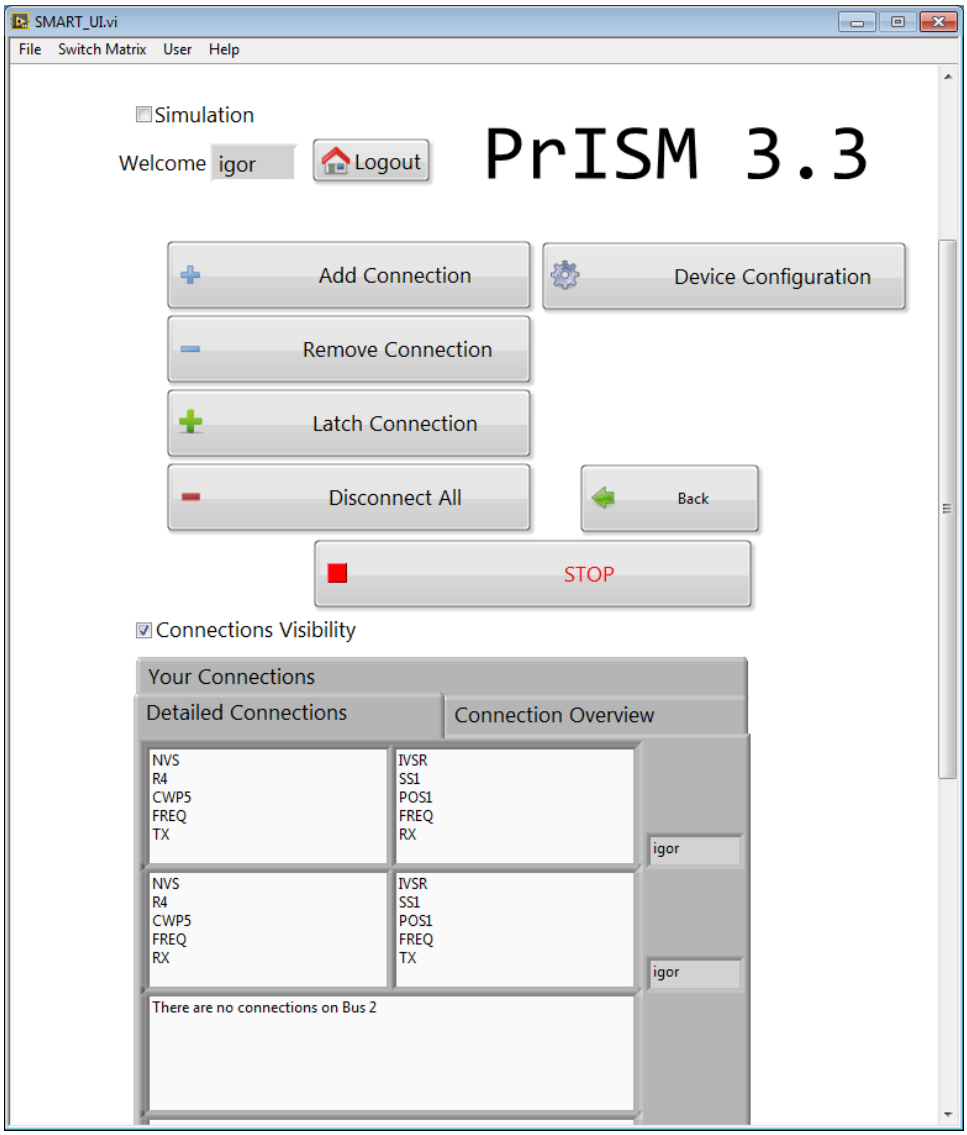


Figure VIII.5. Testing Setup in PrISM

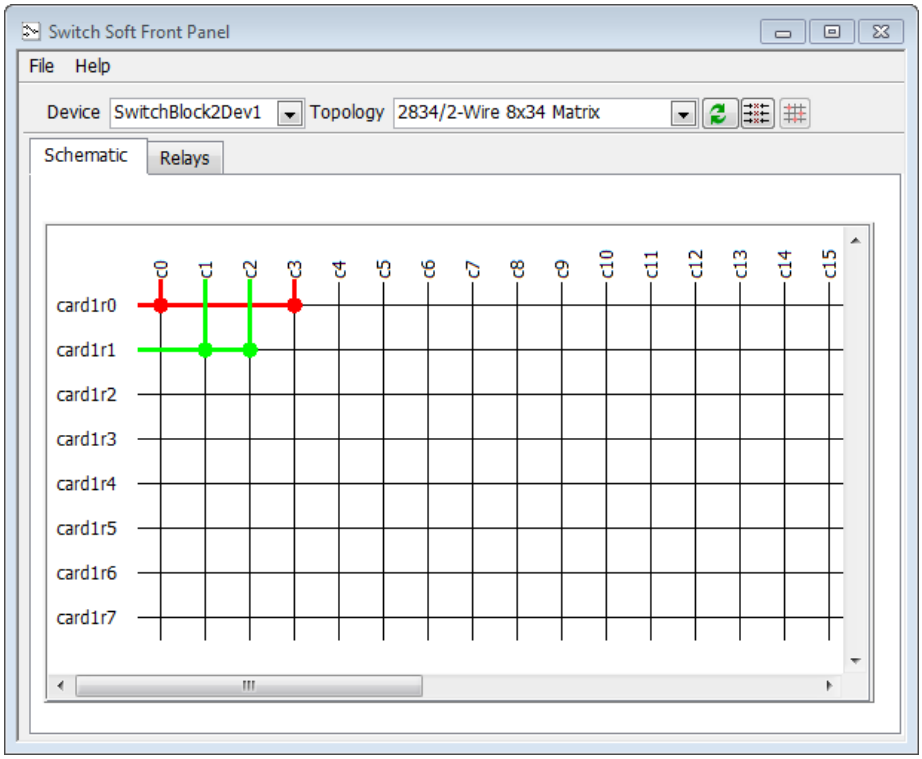


Figure VIII.6. Testing Setup in NI-MAX

Like the previous test results, the results of this round of tests were positive. The one-way delay, PESQ, and PESQ-MOS results can be seen in Appendix E.4. The one-way delay is slightly longer using PrISM, but this difference is statistically insignificant. The PESQ and PESQ-MOS signal quality metrics are slightly higher using PrISM. This can be attributed to Voice 2 having some issues during the tests that didn't include PrISM. The measurement algorithm seemed to have a difficult time with Voice 2. With those results removed from consideration, the signal quality measurements are almost identical with or without PrISM. The results from the VTIMS tests are shown in Appendix F.3 (without PrISM) and F.4 (with PrISM).

IX Conclusions

The results prove that the hardware-software combination in PrISM can operate a switch matrix from a user-friendly GUI, while providing all of the functionality needed from a large switching point system. The project was successfully merged with the thesis project of another UCO student, in order to provide the addition of a testing suite for impairment in communication devices. The addition of the project means that PrISM can perform the tests by being called on by the already developed queued state machine. The implication of this is that any additional modular pieces can be added as long as they are written in the same format to be recognized by PrISM. All of these modular pieces can correspond to their own hardware, all being controlled in parallel from the same graphical user interface.

The university-industry collaboration that drove this project to its completion has proven to be successful in merging these two environments, as has been done in other projects [11] [12] [16]. The hardware/software pairing is fully functional and ready to be used for the current application. The success of this project was achieved via thorough and informative communication by the university counterpart, seeking expertise and feedback from the industry representatives. All hardware benchmarks that were set for the project were met. In addition, the integration of PrISM with communication devices introduced no visible signal deterioration, impairment, or delay.

In order to achieve the previously mentioned streamlined project development between the two entities [17], there is communication underway in order to develop more projects that can use the developed PXI platform. One such project, as previously mentioned, is the possible use of a PXI

controlled virtual transmission impairment measurement set in order to perform line integrity testing. The project can be developed by using compatible programming techniques, in order to develop an embedded VTIMS. The goal is to enhance the functionality of PrISM, as shown in Figure IX.1.

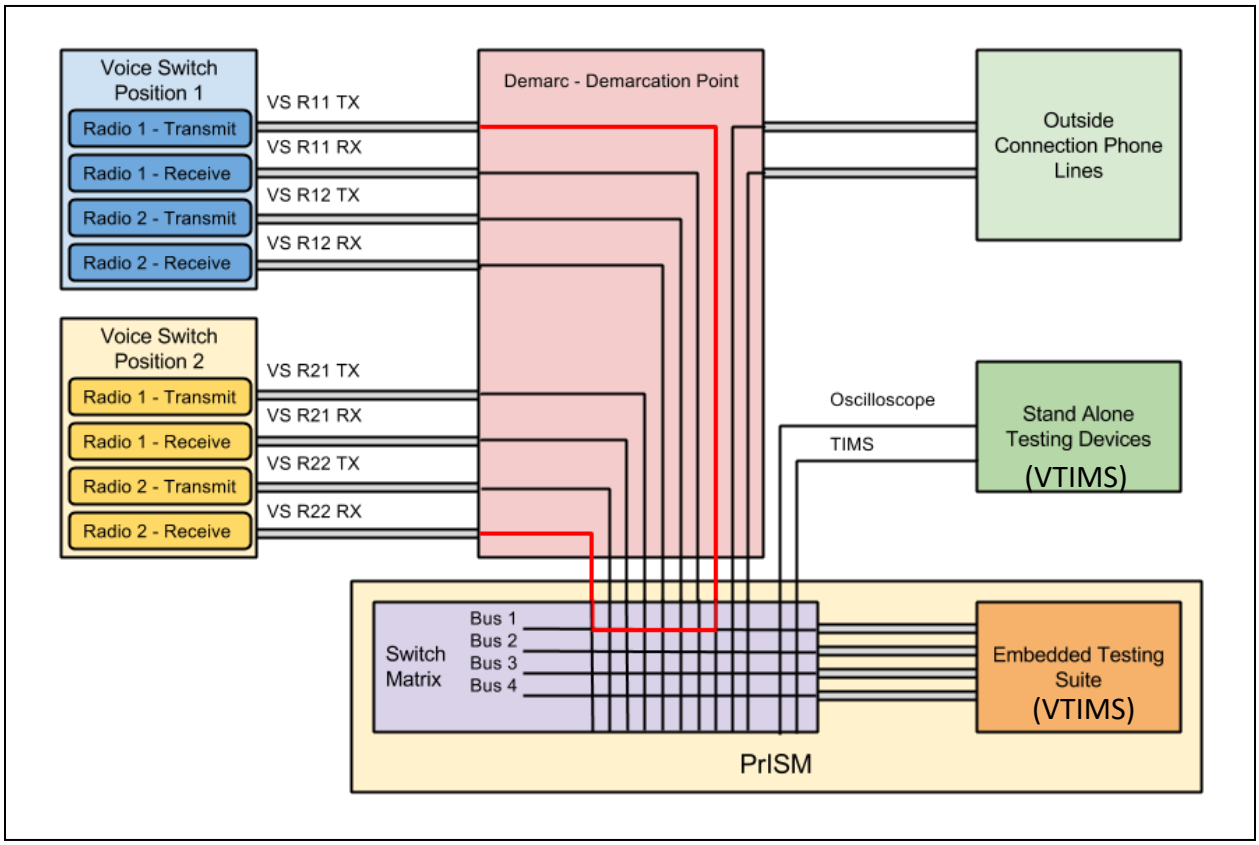


Figure IX.1. OKCET Lab Projected Overview

Further developments on this project will conclude with the completion of the graduate student thesis that the project is centered around. Other possible projects that can be spawned of the basic PrISM software are, but are not limited to:

- Remote control of the switch matrix and testing

- Continued development of a basic login system
- Report generation on user request
- Usage of embedded controllers

The main addition to the current PrISM setup would be remote-control capability. The addition of this functionality would involve another project with its own set of hardware and software to interface PrISM to another end-device (computer, laptop, tablet, etc.). Another set of challenges for this project is the different firewall/login setups for different applications of PrISM. NI has a lot of options and capabilities for remote control of a PXI [35] [36]. The identification of the necessary hardware and functionality was not in the scope of this project, but it can be utilized by another project due to the modular setup of PrISM and the capabilities of integrating other NI hardware.

One drawback from using prepackaged NI hardware and software is the cost. Making this project a more viable commercial product that could be more widely used would require lowering the upfront cost of buying the hardware needed for each setup. Further research into developing an embedded controller and hardware system that can accommodate for the functionality needed can reduce the total cost. Customized hardware solutions can cost a lot less when bought in a large volume compared to the flexible NI packaged products.

All of these future options allow for multiple projects to be spawned by using PrISM as the graphical user interface in this switch matrix setup. This justifies the emphasis that was put on setting up PrISM as an integrated, modular platform.

Bibliography

[1] N. Instruments, "Creating a Large Switch Matrix [White paper]," National Instruments, 2009.

[2] N. Instruments, "Matrix Switch Expansions Guide," National Instruments, 2006.

[3] G. Dimitrakopoulos, Designing Network On-Chip Architectures in the NanoScale Era, Chapman and Hall, 2010, pp. 67-88.

[4] K. Y. Chan, "Monolothic Crossbar MEMS Switch Matrix," *IEEE MTT-S International,* 2008.

[5] R. D. Hoare, "A Near-Optimal Real-Time Hardware Scheduler for Large Cardinality Crossbar Switches," in *SC Conference, Microwave Symposium Digest*, 2006.

[6] N. Instruments, "NI-DAQmx Device Pinouts," 2011. [Online]. Available: http://www.ni.com/white-paper/4053/en/. [Accessed 22 December 2014].

[7] B. Campbel, "Automatic Systems Facilitate Efficient Reed Relay Characterization," *Electronics Industry,* vol. 10, no. 10, pp. 37-39, 1984.

[8] L. Xu, J. Zhang and B. Miedzinski, "New Design of Multi-Contact Reed Relay for Improving Switching Load Capacity," in *Annual Holm Conference of Electrical Contacts*, 1998.

[9] E. Malone and H. Lipson, "Freefirn Fabrication of a Complete Electromechanical Relay," in *Solid Freeform Fabrication Symposium*, 2007.

[10] N. Instruments, "Choosing the Right PXI System Architecture," 2013. [Online]. Available: http://www.ni.com/white-paper/2722/en/.

[11] P. J. de Jongh and C. M. Erasmus, "Industry-directed training and research programmes: The BMI experience," *South African Journal of Science,* vol. 110, no. 11/12, pp. 17-24, 2014.

[12] C. E. Waters, S. Alvine and M. Eble-Hankins, "Industry-Experienced Graduate Student Program: Innovative Collaboration in Architectural Engineering at the University of Nebraska, Lincoln," *Journal of Architectural Engineering,* vol. 18, no. 1, pp. 61-63, 2012.

[13] G. Abramo, C. D'Angelo and F. Di Costa, "University-Industry Research Collaboration: A Model to Asses University Capability," *Higher Education,* vol. 62, no. 2, pp. 163-181, 2011.

[14] G. Watts, "Collaboration Between Academia and Industry is Key o UK Research Success," *BMJ,* vol. 342, no. 2, pp. D1970-d1970, 2011.

[15] T. Behrens and D. Gray, "Unintended Consequences of Cooperative Research: Impact of Industry Sponsorship on Climate for Academic Freedom and Other Graduate Student Outcome," *Research Policy,* vol. 30, no. 2, pp. 179-199, 2001.

[16] O. Lucia, J. M. Burdio, J. M. Acero, L. A. Barragan and J. R. Garcia, "Educational Opportunities Based on the University-Industry synergies in an Open Innovation Framework.," *European Journal of Engineering Education,* vol. 37, no. 1, pp. 15-28, 2012.

[17] C. Nielsen and K. Cappelen, "Exploring the Mechanisms of Knowledge Transfer in University-Industry Collaborations: A Study of Companies, Students, and Researchers.," *Higher Education Quarterly,* vol. 68, no. 4, pp. 375-393, 2014.

[18] M. Wolcott, S. Brown, M. King, D. Ascher-Barnstone, T. Beyeruther and K. Olsen, "Model for Faculy, Student, and Practioner Development in Sustainability Engineering Through an Integrated Design Experience," *Journal of Professional Issues in Engineering Education and Practice,* vol. 137, no. 2, p. 94, 2011.

[19] J. Bruneel, P. D'Este and A. Salter, "Investigating The Factors That Diminish The Barriers to University-Industry Collaboration," *Research Policy,* vol. 39, no. 7, pp. 858-868, 2010.

[20] N. Instruments, "NI-DAQmx Simulated Devices," 2013. [Online]. Available: http://www.ni.com/white-paper/3698/en/. [Accessed 22 December 2014].

[21] N. Instruments, "Using Test Panels in Measurement & Automation Explorer for Devices Supported by NI-DAQmx," 2010. [Online]. Available: http://www.ni.com/white-paper/4638/en/. [Accessed 22 December 2014].

[22] N. Instruments, "Virtual Devices," 2013. [Online]. Available: http://www.ni.com/white-paper/4752/en/. [Accessed 27 December 2014].

[23] N. Instruments, "Tutorial: Sub-VIs," 2008. [Online]. Available: http://www.ni.com/white-paper/7593/en/. [Accessed 27 December 2014].

[24] L. Hao and G. Stitt, "Virtual Finite-State-Machine Architectures for Fast Compilation and Portability," in *IEEE 24t International Conference on Application-Specific Systems, Architectures and Processors*, 2013.

[25] L. Anthony, "LabVIEW Queued State Machine Architecture," National Instruments, 2013.

[26] S. Rein, F. Fitzek and M. Reisslein, "Enabling Improved Speaker Recognition by Voice Quality Estimation," in *IASTED International Conference on Internet and Multimedia Systems and Applications*, 2003.

[27] S. Niranjan, S. Choudbury and J. D. Gibson, "MOSx and Voice Outage Rate in Wireless Communication," in *International Wireless Communications and Mobile Computing Conference*, 2006.

[28] A. L. Bartos and D. J. Nelson, "Voice Capacity Under Quality Constraints for IEEE 802.11a based WLANs," in *Asilomar Conference on Signals, Systems, and Computers*, 2011.

[29] E. U. Warriach and K. Tei, "Fault Detection in Wireless Sensor Networks: A Machine Learning Approach," in *IEEE 16th International Conference on Computational Science and Engineering*, 2013.

[30] A. E. Conway, "Output-Based Method of Applying PESQ to Measure The Perceptual Quality of Framed Speech Signals," in *IEEE Wireless Communcations and Networking Conference*, 2004.

[31] S. Paulsen and T. Uhl, "Quanifying the Sustainability of Reference Signals for the PESQ Algorithm," in *3rd International Conference on Communcation Theory, Reliability, and Quality of Service CTRQ*, 2010.

[32] W. Y. Chan and T. H. Falk, Machine Assessment of Speech Communication Quality, 2012, pp. 587-600.

[33] G. Yi and W. Zhang, "The Perceptual Objective Listening Quality Assessment Algorithm in Telecommunication: Introdcution of ITU-T New Metrics POLQA," in *IEEE Communcations in China*, 2012.

[34] S. Wang, L. Nieto and E. Zielinski, "VoIP QoS Performance Evaluation in a Commercial Environment," in *SPIE- The International Society for Optical Engineering*, 2001.

[35] N. Instruments, "PXI Remote Control and System Expansion," 2017. [Online]. Available: http://www.ni.com/pdf/product-flyers/pxi-remote-control-and-system-expansion.pdf. [Accessed 23 07 2017].

[36] N. Instruments, "Thunderbolt 3 Remote Control of PXI Test Systems," 5 July 2017. [Online]. Available: http://www.ni.com/white-paper/53757/en/. [Accessed 23 July 2017].

APPENDIX A - Hardware

- Computer for interface with LabVIEW and most recent LabVIEW program.
- NI PXIe-1073 PXI Chassis.
- PXI-2800 SwitchBlock Carrier.
- NI 2834 8x34 electromechanical relay card.
- Additional wiring/setup.
- Testing devices.
- Sample end-devices for communication.

APPENDIX B - Configuration files

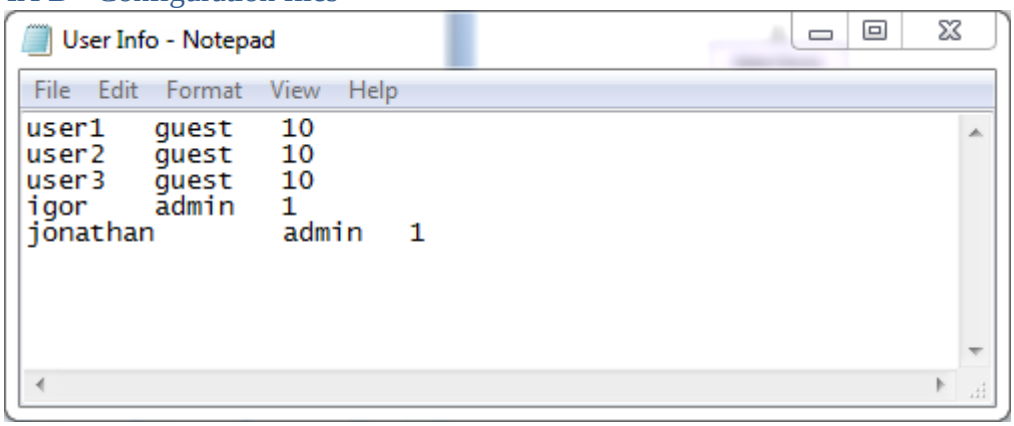


Figure B.1 User Information text file

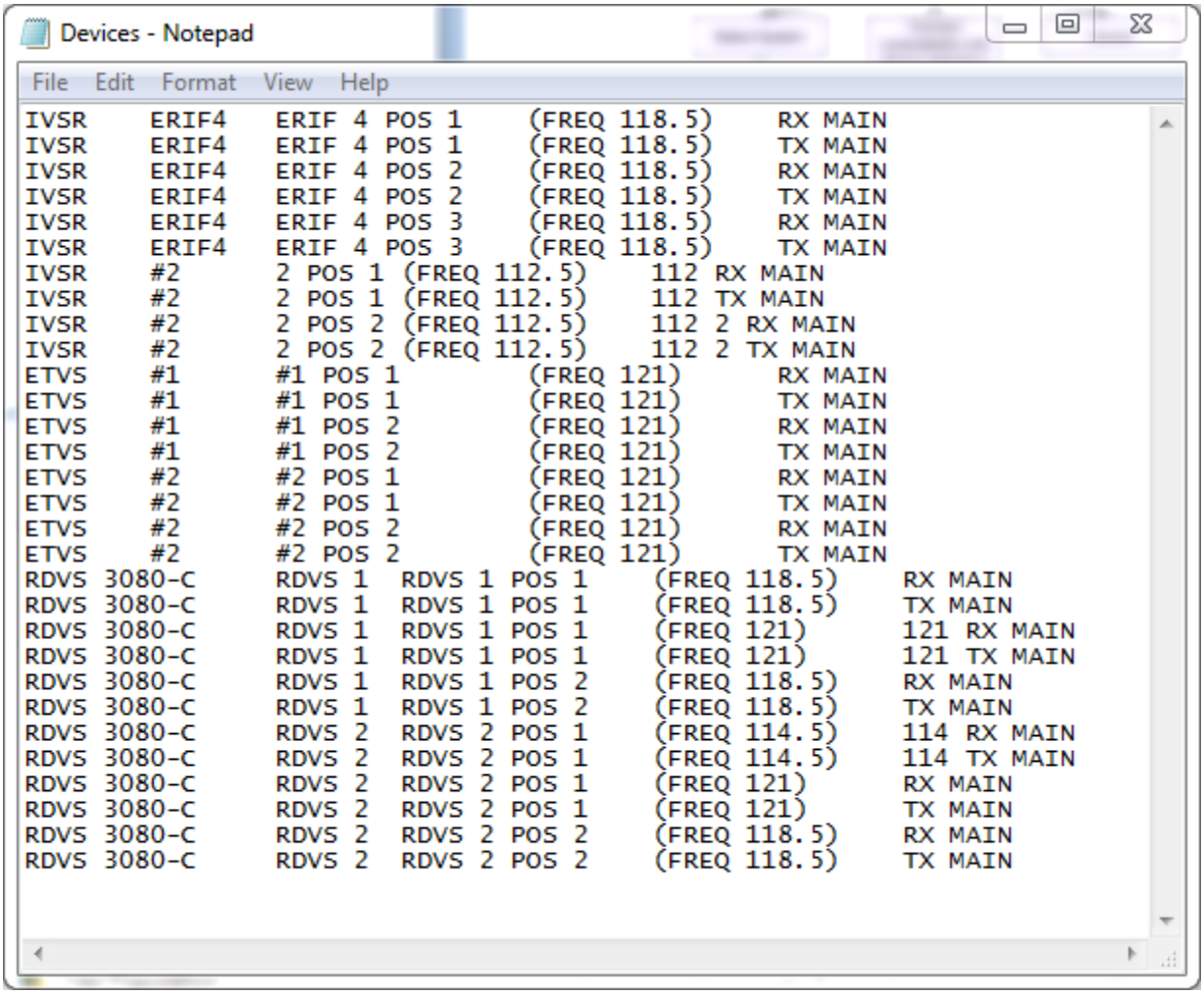
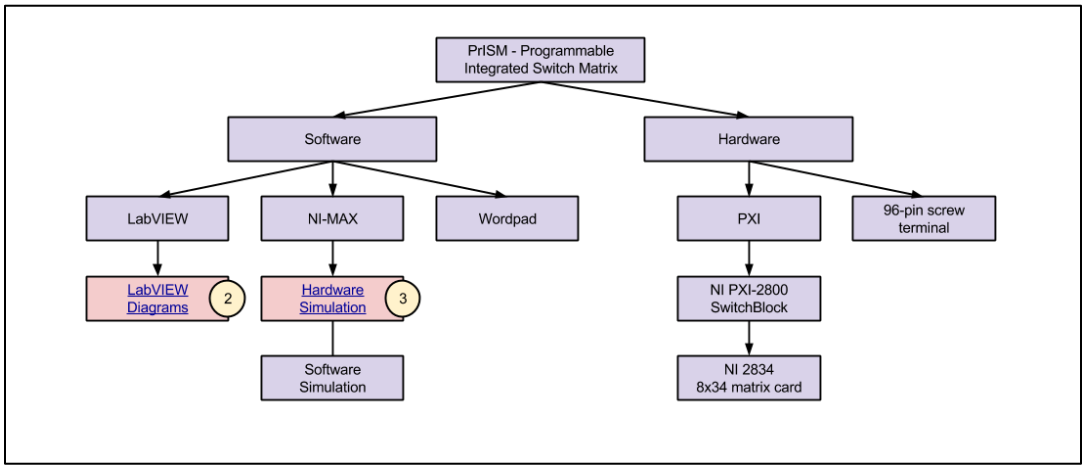
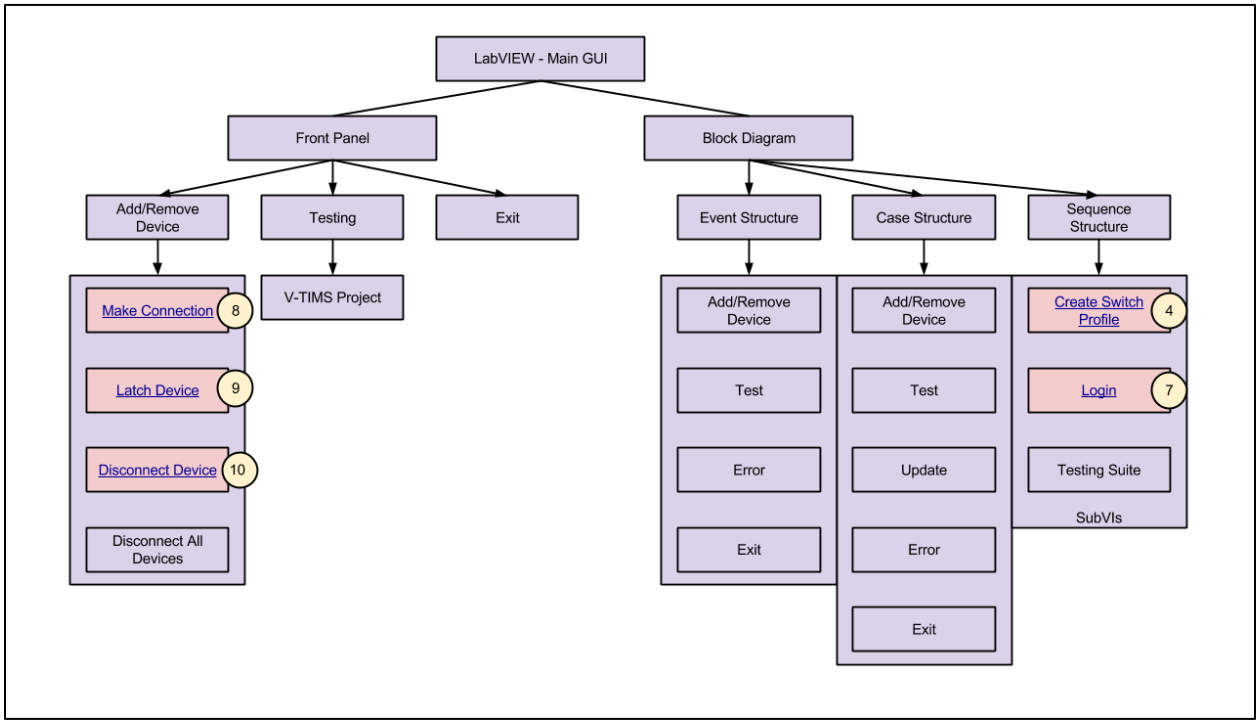
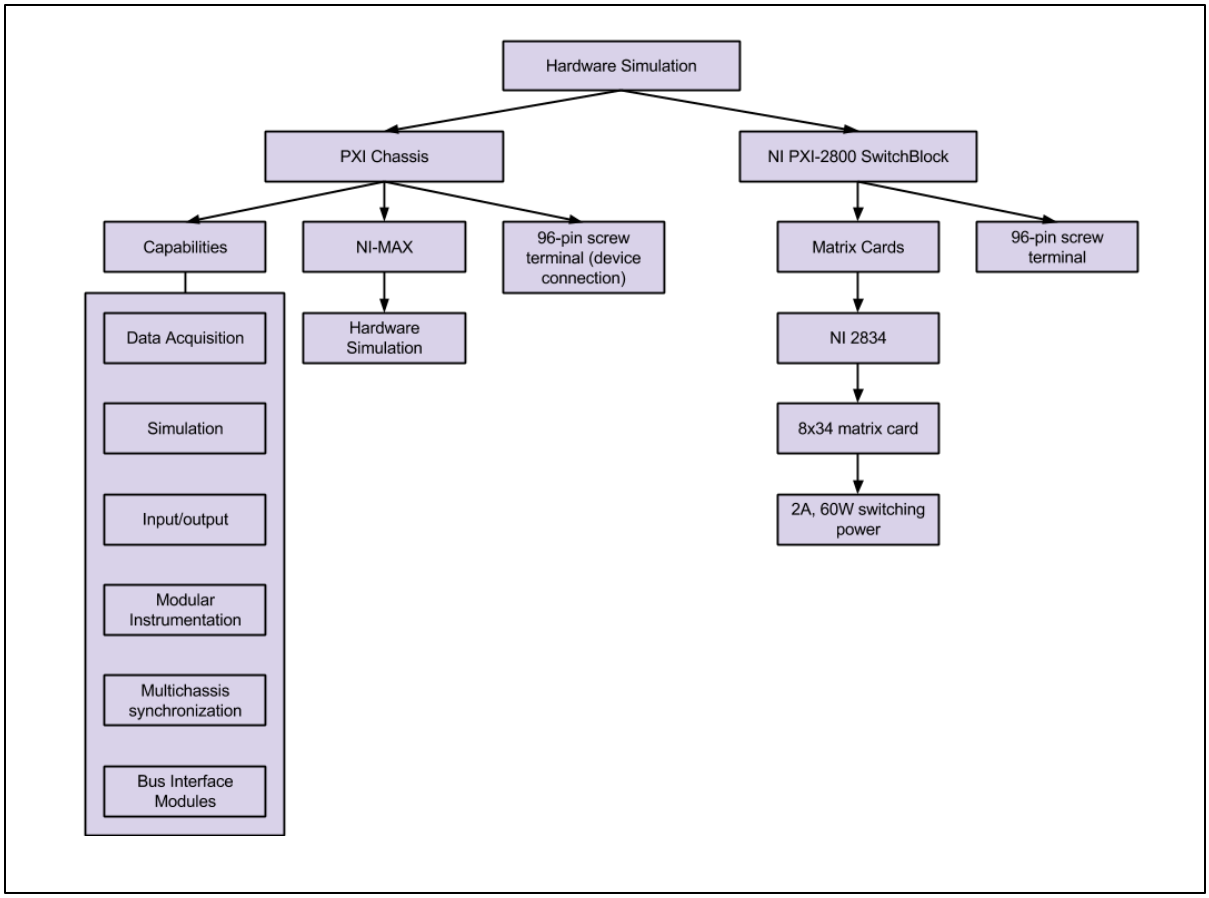


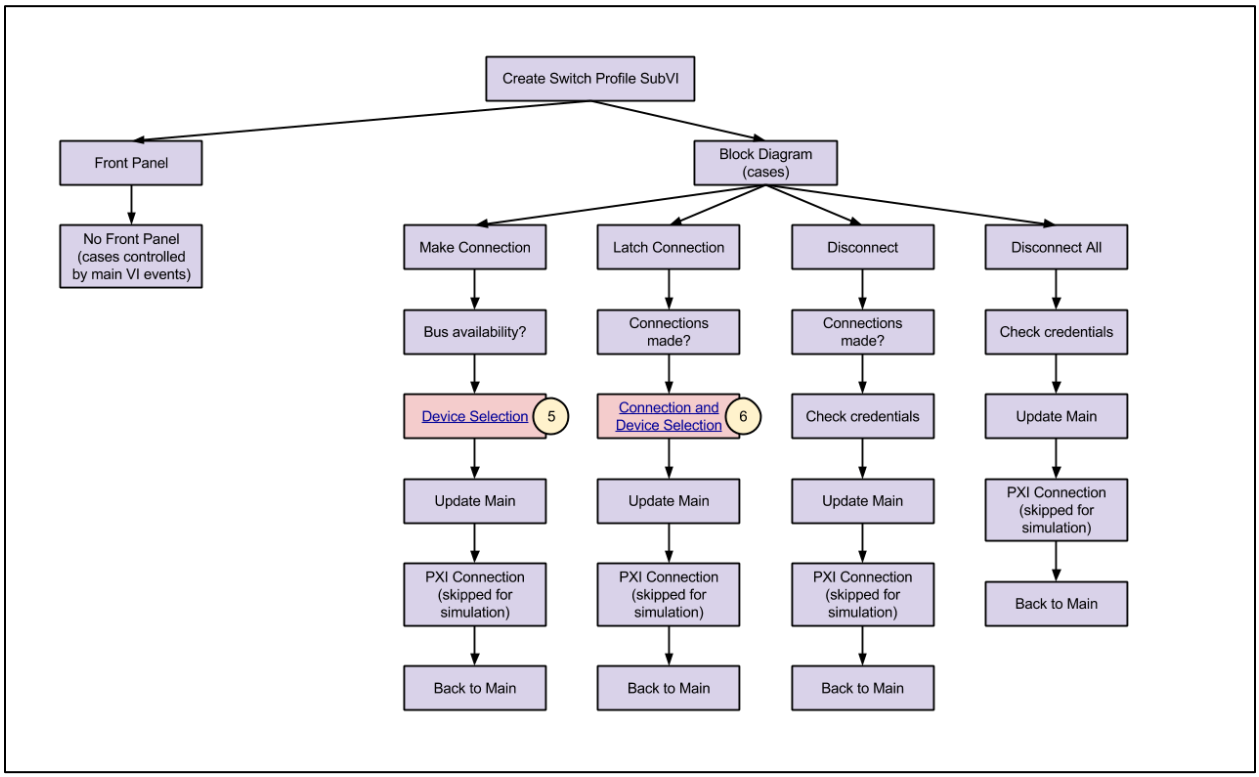
Figure B.2 Device Information text file

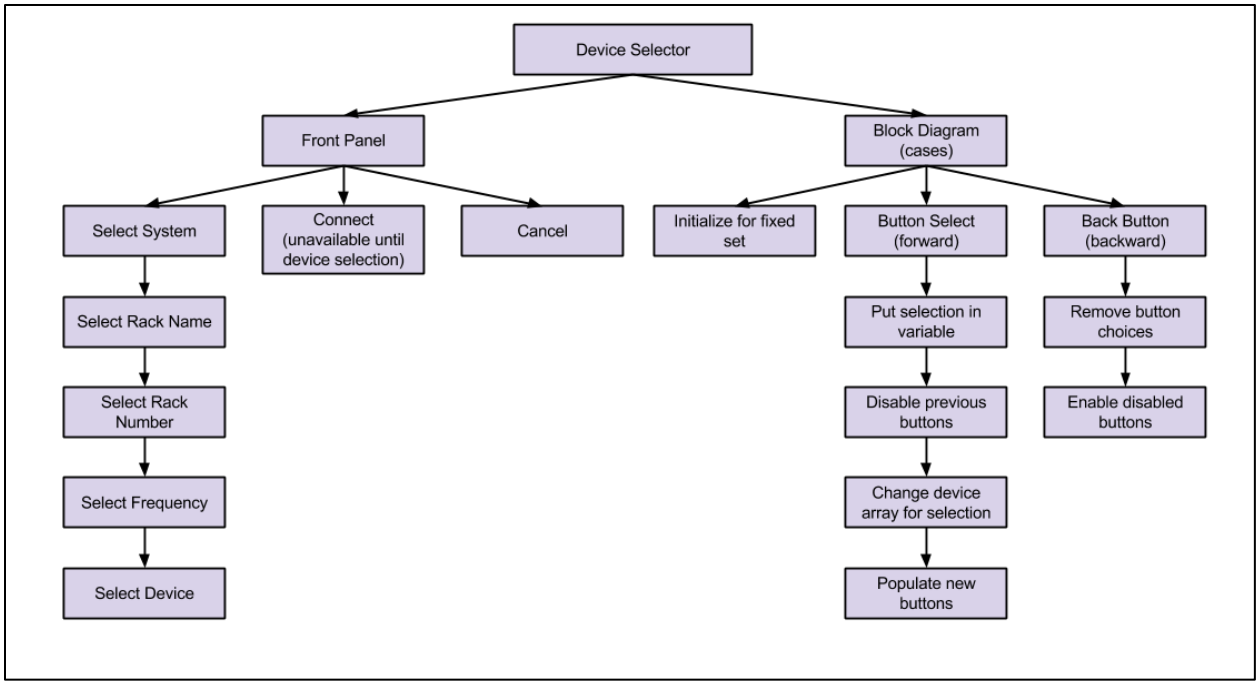
APPENDIX C - System logic pathways

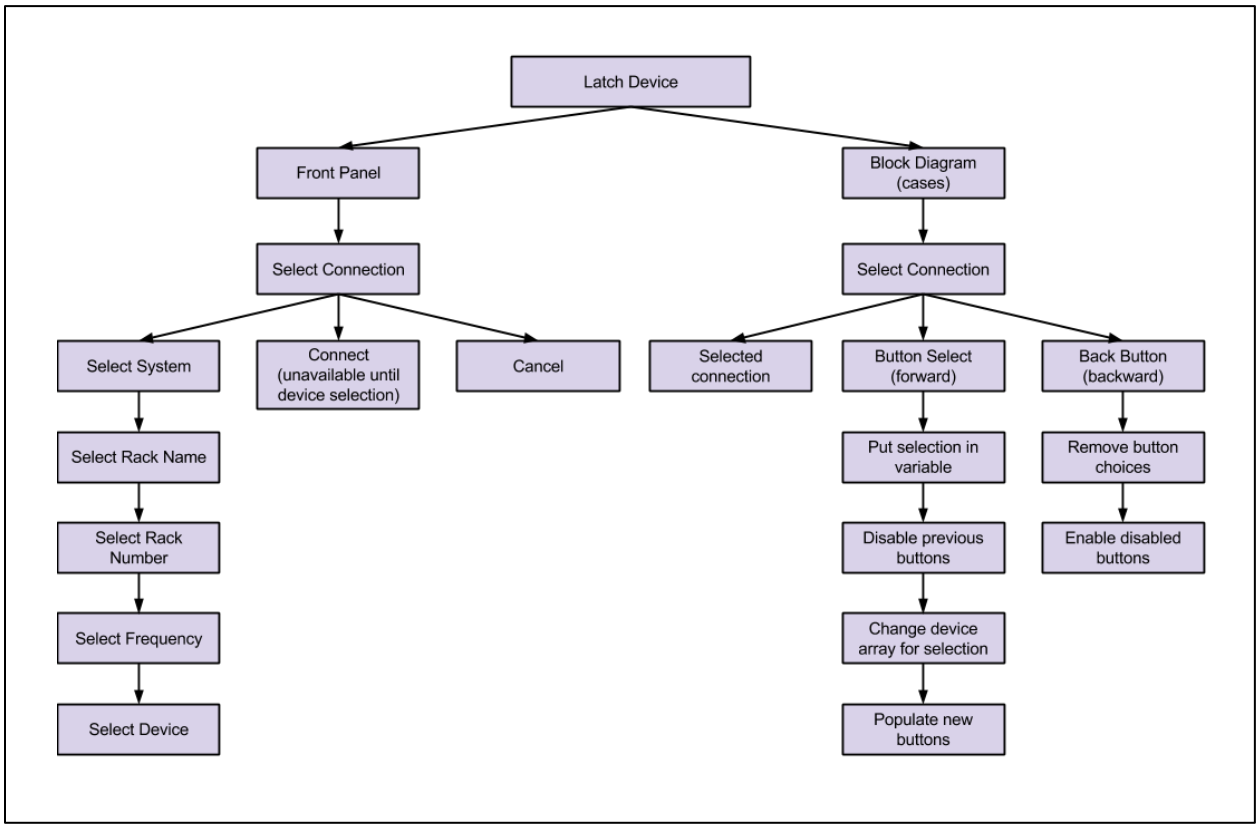





Hardware Simulation
PXI Chassis
NI PXI-2800 SwitchBlock
Capabilities
NI-MAX
96-pin screw terminal (device connection)
Matrix Cards
96-pin screw terminal
Data Acquisition
Simulation
Input/output
Modular Instrumentation
Multichassis synchronization
Bus Interface Modules
Hardware Simulation
NI 2834
8x34 matrix card
2A, 60W switching power


Create Switch Profile SubVI
Front Panel
No Front Panel (cases controlled by main VI events)
Block Diagram (cases)
Make Connection
Latch Connection
Disconnect
Disconnect All
Bus availability?
Connections made?
Connections made?
Check credentials
Device Selection
5
Connection and Device Selection
6
Check credentials
Update Main
Update Main
Update Main
Update Main
PXI Connection (skipped for simulation)
PXI Connection (skipped for simulation)
PXI Connection (skipped for simulation)
PXI Connection (skipped for simulation)
Back to Main
Back to Main
Back to Main
Back to Main


Device Selector
Front Panel
Block Diagram (cases)
Select System
Connect (unavailable until device selection)
Cancel
Initialize for fixed set
Button Select (forward)
Back Button (backward)
Select Rack Name
Select Rack Number
Select Frequency
Select Device
Put selection in variable
Disable previous buttons
Change device array for selection
Populate new buttons
Remove button choices
Enable disabled buttons


Latch Device
Front Panel
Block Diagram (cases)
Select Connection
Select Connection
Select System
Connect (unavailable until device selection)
Cancel
Selected connection
Button Select (forward)
Back Button (backward)
Select Rack Name
Select Rack Number
Select Frequency
Select Device
Put selection in variable
Disable previous buttons
Change device array for selection
Populate new buttons
Remove button choices
Enable disabled buttons

APPENDIX D - Testing Results (Demarc Connection)

1. Connection Test

Connection Test (TIMS 1004 Hz)			
Channel	Connected to:	Attenuation	Test Success
0	1	-0.1 dB	Pass
1	0	-0.1 dB	Pass
2	0	-0.1 dB	Pass
3	0	-0.1 dB	Pass
4	0	-0.1 dB	Pass
5	0	-0.1 dB	Pass
6	0	-0.1 dB	Pass
7	0	-0.1 dB	Pass
8	0	-0.1 dB	Pass
9	0	0.002 dB	Pass
10	0	-0.1 dB	Pass
11	0	-0.1 dB	Pass
12	0	-0.1 dB	Pass
13	0	0.002 dB	Pass
14	0	0.002 dB	Pass
15	0	0.002 dB	Pass
16	0	0.002 dB	Pass
17	0	0.002 dB	Pass
18	0	0.002 dB	Pass
19	0	0.002 dB	Pass
20	0	0.002 dB	Pass
21	0	0.002 dB	Pass
22	0	0.002 dB	Pass
23	0	0.002 dB	Pass
24	0	0.002 dB	Pass

2. Multiple Connection Test

Multiple Connections Test			
Testing Number of Connections	Devices	Devices Fail to Connect	Test Success
2	VTIMS; DMM	None	Pass
3	TIMS; VTIMS; DMM	None	Pass
4	VTIMS; TIMS1; TIMS2; DMM	None	Pass

3. Impairment Tests

Impairment Test (using VTIMS)

Channel	Connected to:	Test(s) Failed	Test Success
0	1	None	Pass
1	0	None	Pass
2	0	None	Pass
3	0	None	Pass
4	0	None	Pass
5	0	None	Pass
6	0	None	Pass
7	0	None	Pass
8	0	None	Pass
9	0	None	Pass
10	0	None	Pass
12	0	None	Pass
18	0	None	Pass
24	0	None	Pass

4. SNR

SNR Ratio	
VTIMS to TIMS	53.3
TIMS to VTIMS	92
VTIMS to VTIMS	61
TIMS to TIMS (same device)	89.9

5. SNR Different Relays

SNR Ratio	
TIMS 1 to TIMS 2 - bus 0	76.5 dBrnc
TIMS 2 to TIMS 1 - bus 1	89.9 dBrnc
TIMS 1 to TIMS 2 - bus 1	76.5 dBrnc
TIMS 2 to TIMS 1 - bus 0	89.9 dBrnc
TIMS 1 to TIMS 2 - bus 3	76.5 dBrnc
TIMS 2 to TIMS 1 - bus 5	89.8 dBrnc

APPENDIX E - Testing Results (Devices Connection)

1. Check Connections (Demarc and VXI system)

Demarc and VXI system	
Test	Result
SS1 RX to SS2 TX (dial tone)	Pass
SS2 RX to SS1 TX (dial tone)	Pass
SS1 RX to SS2 TX (sound check)	Pass
SS2 RX to SS1 TX (sound check)	Pass

Connection: Made

2. Passing frequency at 704 Hz

TIMS tests			
TX	RX	Freq shift	Result
SS1 TIMS transmitting at -10dB	SS2 TIMS receiving at -26dB	None	Pass
SS2 TIMS transmitting at -10dB	SS1 TIMS receiving at -26dB	None	Pass
SS1 VTIMS transmitting at 0dB	SS2 TIMS receiving at -16dB	None	Pass
SS2 VTIMS transmitting at 0dB	SS1 TIMS receiving at -16dB	None	Pass
SS1 TIMS transmitting at -10dB	SS2 VTIMS receiving at -26dB	None	Pass
SS2 TIMS transmitting at -10dB	SS1 VTIMS receiving at -26dB	None	Pass

3. Delay and voice quality test

Demarc and VXI system with and without PrISM			
Test	Without PrISM	With PrISM	Result
OWD (one way delay) 1	3.9 ms	4.0 ms	
OWD 2	4.0 ms	4.0 ms	
OWD 3	3.9 ms	4.0 ms	
OWD Average	**3.9 ms**	**4 ms**	**Pass**
PESQ 1	3.67	3.42	
PESQ 2	3.67	3.87	
PESQ 3	3.67	3.67	
PESQ Average	**3.67**	**3.69**	**Pass**
PSQM MOS 1	4.61	4.53	
PSQM MOS 2	4.61	4.69	

PSQM MOS 3	4.61	4.61	
PSQM MOS Average	**4.61**	**4.61**	**Pass**

4. Delay and voice quality tests (Demarc Only)

Demarc only with and without PrISM			
Test	Without PrISM	With PrISM	Result
OWD (one way delay) 1	60.4 ms	60.9 ms	
OWD 2	59.7 ms	59.8 ms	
OWD 3	59.1 ms	59.2 ms	
OWD Average	**59.7 ms**	**60.0 ms**	**Pass**
PESQ Voice 1	4.02	4.02	
PESQ Voice 2	3.59	4.19	
PESQ Voice 3	4.12	4.11	
PESQ Voice 4	4.11	4.11	
PESQ Voice 5	4.20	4.19	
PESQ Voice 6	4.16	4.17	
PESQ Average	**4.03**	**4.13**	**Pass**
PSQM MOS 1	4.86	4.86	
PSQM MOS 2	4.65	4.87	
PSQM MOS 3	4.86	4.86	
PSQM MOS 4	4.90	4.90	
PSQM MOS 5	4.92	4.91	
PSQM MOS 6	4.91	4.91	
PSQM MOS Average	**4.85**	**4.89**	**Pass**

APPENDIX F - VTIMS Report

1) Test 1 Report 1 – VXI System

INTERIM TECHNICAL PERFORMANCE RECORD — **Fiber Optic Transmission System (FOTS) Analog VF Services**

FACILITY	DATES				SUPERVISOR'S SIGNATURE
ATC	FROM	*NA*	To	*NA*	

LOCATION *OKC, OK*

Date	Time	Circuit ID	Sending Facility				Receiving Facility				Initials
		yes	*Traffic Facility*				*Traffic Facility*				
		Configuration: End-to-End / Looped / Usage: All	Line Test Set-Up (1004 Hz)								
		No / No / No	Send Point: *0*	TLP	*-13*	dBm	Receive Point: *0.00*	TLP	*-13.0*	dBm	

		1004 Hz Net Loss/ Frequency Shift		Attenuation Distortion 304 to 3004 Hz		3 Tone Slope Attenuation Distortion Or Attenuation Distortion 404 to 2804 Hz		Signal-to-C-notched Noise Ratio)	Remarks
				304 Hz – 3004 Hz		404 Hz – 2804 Hz			
		Freq. Shift	Level (dB)	Minimum Level (dB)	Maximum Level(dB)	Minimum Level (dB)	Maximum Level (dB)	Level (dB)	
Nominal		0 Hz	0.0 dB	1004 Hz Net Loss	1004 Hz Net Loss	1004 Hz Net Loss	1004 Hz Net Loss	≥ 34 dB	
Minimum		-1Hz	-1.5 dB		-1.0 dB		- 1.0 dB	> 32 dB	
Maximum		+1Hz	+1.5 dB	+5.0 dB		+2.0 dB			
6/8/20 15	*10:20 AM*	*0.00*	*0.00*	*0.44*	*-0.03*	*0.37*	*-0.03*	*49.5*	

Receive Level

Freq (Hz)	304	404	504	604	704	804	904	1004	1104	1204	1304	1404	1504	1604	Remarks	
Attenuation dBm	*-0.02*	*-0.03*	*-0.03*	*-0.03*	*-0.03*	*-0.02*	*-0.01*	*0.00*	*0.01*	*0.02*	*0.04*	*0.05*	*0.07*	*0.08*		
Attenuation dBm																
Attenuation dBm																
Freq (Hz)	1704	1804	1904	2004	2104	2204	2304	2404	2504	2604	2704	2804	2904	3004	Remarks	
Attenuation dBm	*0.10*	*0.12*	*0.14*	*0.17*	*0.19*	*0.21*	*0.24*	*0.26*	*0.29*	*0.32*	*0.35*	*0.38*	*0.41*	*0.44*		
Attenuation dBm																
Attenuation dBm																

2) Test 1 Report 2

INTERIM TECHNICAL PERFORMANCE RECORD — **Fiber Optic Transmission System (FOTS) Analog Data Services**

FACILITY	DATES				SUPERVISOR'S SIGNATURE
ATC	FROM	*NA*	TO	*NA*	

LOCATIONOKC, OK

Date	Time	Circuit ID	Sending Facility	Receiving Facility										Initials
		yes	*Traffic Facility*	*Traffic Facility*										
		Configuration		Usage	Line Test Set-Up (1004 Hz)									
		End-to-End	Looped	Data	Send Point				Receive Point					
		No	*No*	*No*	*0*	TLP	*-13*	dBm	*0.00*	TLP	*-13.0*	dBm		

Date	Time	Impulse Noise	Envelope delay distortion	Phase Jitter		Intermodulation Distortion		Remarks	Initials
		At Threshold Noted (dBmc0)	804 Hz – 2604 Hz	4–300 Hz	20–300 Hz	Second Order	Third Order		
Nominal		≤ 15 counts in 15 Min. at 65 dBmC0	≤650 µsec	≤8 °	≤3 °	≥ 46 db	≥ 49 db		
Minimum						> 45 db	> 48 db		
Maximum		≤ 15 counts in 15 Min. at 67 dBmC0	≤ 700 µsec	<9 °	<4 °				
6/8/2015	*10:20 AM*	0	6	0.6	0.6	64.9	66.2		

Envelope Delay Distortion

Freq (Hz)	750	906	1063	1219	1375	1531	1688	1845	2000	2156	2313	2469	2625	2781	Remarks	
Delay (µ sec)	*-0*	*-1*	*2*	*-4*	*2*	*0*	*-0*	*0*	*-2*	*0*	*-0*	*0*	*-1*	*0*		
Delay (µ sec)																
Delay (µ sec)																

3) Test 2 Report 1 – No VXI System

INTERIM TECHNICAL PERFORMANCE RECORD — **Fiber Optic Transmission System (FOTS) Analog VF Services**

FACILITY	DATES				SUPERVISOR'S SIGNATURE
OKC, OK	FROM	*Traffic Facility*	To	*Τραφφικ Φαχιλιτψ*	

LOCATION *ATC*

Date	Time	Circuit ID			Sending Facility				Receiving Facility				Initials
		ID			*Traffic Facility*				*Traffic Facility*				
		Configuration		Usage	Line Test Set-Up (1004 Hz)								
		End-to-End	Looped	All	Send Point				Receive Point				
		Yes	No	Yes	*0*	TLP	*-13*	dBm	*0.00*	TLP	*-28.8*	dBm	

		1004 Hz Net Loss/ Frequency Shift		Attenuation Distortion 304 to 3004 Hz		3 Tone Slope Attenuation Distortion Or Attenuation Distortion 404 to 2804 Hz		Signal-to-C-notched Noise Ratio)	Remarks
				304 Hz – 3004 Hz		404 Hz – 2804 Hz			
		Freq. Shift	Level (dB)	Minimum Level (dB)	Maximum Level(dB)	Minimum Level (dB)	Maximum Level (dB)	Level (dB)	
Nominal		0 Hz	0.0 dB	1004 Hz Net Loss	1004 Hz Net Loss	1004 Hz Net Loss	1004 Hz Net Loss	≥ .34 dB	
Minimum		-1Hz	-1.5 dB		-1.0 dB		- 1.0 dB	≥ .32 dB	
Maximum		+1Hz	+1.5 dB	+5.0 dB		+2.0 dB			
10/14/ 2017	*11:17 AM*	*0.00*	*15.80*	*16.20*	*15.80*	*16.20*	*15.80*	*26.5*	

Receive Level

Freq (Hz)	304	404	504	604	704	804	904	1004	1104	1204	1304	1404	1504	1604	Remarks	
Attenuation dBm	*16.00*	*15.90*	*15.90*	*15.90*	*15.90*	*15.80*	*15.80*	*15.80*	*15.90*	*15.80*	*15.80*	*15.80*	*15.80*	*15.80*		
Attenuation dBm																
Attenuation dBm																
Freq (Hz)	1704	1804	1904	2004	2104	2204	2304	2404	2504	2604	2704	2804	2904	3004	Remarks	
Attenuation dBm	*15.80*	*15.80*	*15.80*	*15.80*	*15.80*	*15.90*	*15.90*	*15.90*	*16.00*	*16.00*	*16.00*	*16.10*	*16.10*	*16.20*		
Attenuation dBm																
Attenuation dBm																

4) Test 2 Report 2

INTERIM TECHNICAL PERFORMANCE RECORD																Fiber Optic Transmission System (FOTS) Analog VF Services	
FACILITY							DATES									SUPERVISOR'S SIGNATURE	
OKC, OK							FROM	10/14/2017		To	10/14/2017						
LOCATION ATC																	
Date	Time	Circuit ID					Sending Facility						Receiving Facility				Initials
		ID					Traffic Facility						Traffic Facility				
		Configuration			Usage		Line Test Set-Up (1004 Hz)										
		End-to-End		Looped		All	Send Point						Receive Point				
		Yes		No		Yes	0	TLP	-13	dBm			0.00	TLP	-28.9	dBm	
		1004 Hz Net Loss/ Frequency Shift		Attenuation Distortion 304 to 3004 Hz			3 Tone Slope Attenuation Distortion Or Attenuation Distortion 404 to 2804 Hz				Signal-to-C-notched Noise Ratio)		Remarks				
		Freq. Shift	Level (dB)	304 Hz – 3004 Hz Minimum Level (dB)	304 Hz – 3004 Hz Maximum Level(dB)		404 Hz – 2804 Hz Minimum Level (dB)		404 Hz – 2804 Hz Maximum Level (dB)		Level (dB)						
Nominal		0 Hz	0.0 dB	1004 Hz Net Loss	1004 Hz Net Loss		1004 Hz Net Loss		1004 Hz Net Loss		≥ 34 dB						
Minimum		-1Hz	-1.5 dB		-1.0 dB				- 1.0 dB		≥ 32 dB						
Maximum		+1Hz	+1.5 dB	+5.0 dB			+2.0 dB										
10/14/2017	12:15 PM	0.00	15.90	16.20	15.80		16.30		15.80		32.3						

Receive Level																	
Freq (Hz)	304	404	504	604	704	804	904	1004	1104	1204	1304	1404	1504	1604	Remarks		
Attenuation dBm	16.10	16.00	15.90	15.90	15.90	15.80	15.80	15.80	15.90	15.90	15.90	15.80	15.80	15.80			
Attenuation dBm																	
Attenuation dBm																	
Freq (Hz)	1704	1804	1904	2004	2104	2204	2304	2404	2504	2604	2704	2804	2904	3004	Remarks		
Attenuation dBm	15.80	15.90	15.90	15.80	15.80	15.90	15.90	16.00	16.00	16.10	16.10	16.10	16.20	16.20			
Attenuation dBm																	
Attenuation dBm																	